The Small Computer Magazine

kilobaud

T.M.

Understandable for beginners . . . interesting for experts

September 1978 / Issue #21 / $2.00 / DM 8 / Sfr 8 / Ffr 18 / Sweden Kr 21 / UK £2

40K Bytes RAM Memory
1,200,000 Bytes Disk Storage
Desk with laminated plastic surface
DOS and BASIC with random and sequential files
TERMINAL—Upper-Lower case and full control character decoding

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216

#1

WE'RE FIRST IN TWO... AND WE THANK YOU.

We are what our friends, you the ComputerLand customer, make us.

You support growth. And ComputerLand expands. There are stores across the nation serving you.

You're quality oriented. That's why your ComputerLand store has the widest selection of the finest computer systems and software anywhere.

You have high professional standards. These set the guidelines for your ComputerLand staff . . . experts who can analyze and fill needs at all levels.

You settle for nothing less than complete solutions. That's what gives your store national recognition for its products and service.

In short, you give direction for growth. This is what has made your store #1, in just two short years.

COME TO THE PARTY

And, we're hosting a gigantic birthday party . . . To celebrate our shared success. On September 23, all day. At your local ComputerLand store.

Bring your family, your friends. And join in the fun.

It's our way of saying thanks . . . for yesterday, today and tomorrow.

ComputerLand™

14400 Catalina Street, San Leandro, CA 94577 (415) 895-9363

AL: Huntsville
CA: Dublin, El Cerrito, Hayward, Inglewood, Lawndale, Los Altos, Mission Viejo, San Bernardino, San Diego, San Francisco, San Jose, San Mateo, Santa Rosa, Thousand Oaks, Tustin, Walnut Creek
CO: Colorado Springs, Denver
CT: Fairfield
DE: Newark
FL: Boca Raton, Ft. Lauderdale
GA: Atlanta
HI: Honolulu
IL: Arlington Heights, Downers Grove, Niles, Oak Lawn, Peoria
IN: Indianapolis
KY: Louisville
MD: Rockville
MI: Detroit, Grand Rapids
MN: Minneapolis
NH: Nashua
NJ: Cherry Hill, Morristown, Paramus
NY: Buffalo, Ithaca
NC: Charlotte
OH: Cleveland
OR: Portland
PA: Harrisburg
TX: Austin, Dallas, Houston
WA: Bellevue, Federal Way, Tacoma
WI: Madison
INT'L: Sydney, Australia; Winnipeg, Canada

The OSI 48 Line BUS

OHIO SCIENTIFIC offers the broadest line of BUS compatible microcomputer boards. This line includes several new and exciting products which are not available anywhere else, such as a three processor CPU board, dual port memories and a multiprocessing CPU expander.

OHIO SCIENTIFIC has delivered approximately 100,000 boards based on our 48 line BUS and is now delivering thousands per week in 17 models of computers and dozens of accessories.

OHIO SCIENTIFIC'S BUS design incorporates high band width, high density and mass production technology to achieve a truly remarkable performance to cost ratio.

Here is just a sampling of the many OSI 48 BUS compatible boards available for the systems user, prototyper, OEM user and experimenter.

Product Description	Special Features	Power Supply Voltages Req'd	Board & Doc. Part #	Board & Doc. Price	Assembled Product Part #	Assembled Product Price
CPU						
• Challenger II CPU BASIC-in-ROM 6502 based CPU with serial I/O 4K RAM, machine code monitor	• Can use four 2716 EPROMS instead of BASIC or can be configured for disk	+5/−9	500	39.00	C2-0	298.00
• Challenger III CPU has 6502A, 6800 and Z80 micros, RS-232 serial port, machine code monitor	• 1 megabyte memory manager, software programmable vectors	+5/−9	510	NA	C3-0	490.00
• 560Z multi-processing CPU expander runs PDP-8, Z80 and 8080 code	• Runs concurrently with another OSI CPU	+5/−9	560Z	125.00	NA	NA
RAM						
• 16K static RAM (Ultra low power)	• 215NS access time automatic power down standby mode	+5/+12/−9	520	35.00	CM-3	498.00
• 8K static RAM (low cost)	• Expandable to 16K	+5	—	—	CM-7	198.00
• 16K static RAM (low cost)	• Can be expanded to dual port operation	+5	525	35.00	CM-8	339.00
• 24K static RAM (high density)	• 20 address bits	+5	527	35.00	CM-9	NA
• 4K static RAM (2102 based)	• Can be populated for 4K by 12 bits	+5	420	35.00	CM-2	125.00
• 16K dynamic (ultra low cost)	• Uses 4027 RAMS	+5/+12/−9	530	NA	CM-4	249.00
• 32K dynamic	• 20 address bits	+5/+12/−9	530	NA	CM-5	698.00
• 48K dynamic (high density)	• 20 address bits	+5/+12/−9	530	NA	CM-6	990.00
EPROM Boards						
• 8K 6834 EPROM board	• 16 line parallel port and on board programmer	+5/−9	450	35.00	NA	NA
• 4K 1702A EPROM board	• 16 line parallel port	+5/−9	455	35.00	NA	NA
I/O Boards						
• Audio Cassette interface Kansas City standard 300 baud	• Expandable to CA-7C	+5/−9	430	35.00	CA-6C	99.00
• RS-232 port board	• Expandable to CA-7S	+5/−9	430	35.00	CA-6S	99.00
• Combination audio cassette two 8 bit DACs, one fast A/D and 8 channel input mux	• Also Features 8 parallel I/O lines	+5/−9	430	35.00	CA-7C	399.00
• Combination RS-232 two 8 bit DACs, one fast A/D and 8 channel input mux	• Also features 8 parallel I/O lines	+5/−9	430	35.00	CA-7S	399.00
• 32 by 32 character video display interface	• Keyboard input port	+5/−9	440	35.00	NA	NA
• 32 by 64 character video display interface	• Upper/lower case graphics and keyboard port	+5	540	NA	CA-11	249.00
• 16 port serial board RS-232 and/or high speed synchronous	• 75 to 19,200 baud and 250K and 500K bit rates individually strappable	+5/−9	550	35.00	CA-10X	200.00 to 900.00
• Parallel (Centronics) Line Printer Interface	• With cable	+5/−9	470	NA	CA-9	249.00
• 96 Line Remote Parallel Interface	• Interface "Front End" remotable via 16 pin ribbon cable	+5	—	—	CA-12	249.00
• Voice I/O board with Votrax* module	• Fully assembled voice output, experimental voice input	+5/−9	—	—	CA-14	525.00
DISKS						
• Single 8" floppy disk, 250 Kbytes storage	• Complete with operating system software and disk BASIC	+5/−9	470	NA	CD-1P	790.00
• Dual 8" floppy disk, 500 Kbytes storage	• Complete with operating system software and disk BASIC	+5/−9	470	NA	CD-2P	1390.00
• 74 Million byte Winchester disk and interface	• Complete with OS-65U operating system	+5/−9	—	—	CD-74	6000.00
OTHER						
• 8 slot backplane board with connectors	• Can be daisy-chained to n-slots	—	580	39.00	NA	NA
• Prototyping board	• Handles over 40 16 pin IC's	—	495	29.00	—	—
• Card Extender	• With connectors	—	498	29.00	—	—

For more information, contact your local OHIO SCIENTIFIC Dealer or the factory at (216) 562-3101

OHIO SCIENTIFIC

1333 S. Chillicothe Road • Aurora, Ohio 44202

PUBLISHER'S REMARKS

Wayne Green

How Was NCC?

It beats me why microcomputer firms exhibit at that show. That should be no news to anyone who has been reading my editorials for any length of time. NCC is aimed at the big-frame people and appears to be an enormous waste of time and money for microcomputer people.

The microcomputer exhibits seemed about 100 miles away from the Big Show . . . or at least so it appeared when you got out in the hot sun and waited for the bus. Once you were aboard, the bus seemed to be driving for hours all the way to the opposite end of Disneyland from the Convention Center. In the Disneyland hotel the crowds were OK at the microcomputer show, but just about every exhibitor I talked with had the same observation: These people *aren't* our customers.

The brutal fact is that the people who go to NCC are, for the most part, from ITT, Ma Bell, U.S. Steel and the major banks. There are spies from IBM, Data General, DEC and all of the other Big Boys looking to see what is happening . . . and how their firms can counter it. Most of the micro exhibitors needed to be seen by computer store people . . . and they were not there.

We displayed the first samples of our new software cassettes and generated a lot of interest. Since we weren't supposed to sell anything at the show or even take orders, we naturally threw out the innumerable requests for shipment of the programs and some 2000 requests for subscriptions to *Kilobaud*. You believe that, right? Wouldn't you know that a couple of Japanese would take pictures of the software packages? I'll tell you what, Japan, I'll bet we sell more programs in Japan than you do. Just watch us.

Atlanta Computerfest

It's already getting a little difficult to remember Atlanta as the shows of the past months blend into a sort of blur. It started in April with Percomp at Long Beach. Whether the publicity was lacking, the ads didn't reach anyone or Los Angeles is a stiff for computer shows, the fact is that exhibitors at Percomp were very lonely, often outnumbering the attendees. Even though the perpetrators of this show did not ask me to speak, I did get together two sessions on how to write ads and spec sheets for the exhibitors. I would have worked up a session for the attendees, but I didn't want to talk to myself.

My talks were judged helpful, and they will be scheduled into the coming Dallas show in September. I think we'll charge $100 a head for the workshop and limit it to 20 so I can give individual criticism to ads and other literature. It will be worth it.

After Percomp my next show was Birmingham, where they had a hamfest/computerfest. The computer turnout was good here, and interest was high. I talked on software to a packed room . . . I like that. This was in mid-May.

A few days later Sherry and I headed for Europe to attend the Euro-Micro show in Paris. We stopped off in London for a couple of days and visited a newly opened Byte shop and rapped with John Marshall of Nascom, a new manufacturer of microcomputers in the UK. The next stop was Paris for the show. I was on the program to talk about the development of microcomputing in the U.S., and my ego grew to hitherto unknown dimensions when the hall filled up and a great many people had to be turned away.

We sold a lot of subscriptions at our booth, so the show was a definite success.

From Paris we flew to Zurich and then drove down to Geneva to visit the ITU for a day and find out if the staff there had any suggestions on how to counter the Black Bloc, a group of 44 African countries that have been voting together in recent years, upsetting frequency allocations. Unless something can be done about this situation, there is a possibility that amateur radio might be disenfranchised next year. The ITU staff had little to say that was encouraging.

From Geneva we drove around Switzerland for a day, lunching in Liechtenstein, dropping in on Austria and Germany, and finally returning to Zurich for the flight home.

After one day at home we were off again, this time to Atlanta for the hamfest/computerfest. The hamfest did fine, but the computerfest was not as lively as last year. My talk on software and the opportunities opening in microcomputing to make money was well received.

From there we went directly to Anaheim for NCC, where my talk on software was greeted with a lot of interest and many darned good questions. After NCC we flew up to San Francisco for a day to visit John Peers and Lomac . . . and with Adam, John's computer. While you still have to do a lot of detail work to put a program into Adam, there is no question in my mind that it is far simpler to work with than any other language I've seen so far. If it weren't for the $40K price and my built-in Yankee thrift, I'd probably have walked out with an Adam on order, just for the fun of working with it.

Just a few minutes working with the Lomac Adam system indicates there is no question that it is time for the computer industry to do some heavy thinking about BASIC and the other languages. It has been proven that it is possible to have a much simpler pro-

The Microcomputer Expo in Paris did things up in first-class fashion. The only problem was that the place filled up—including the aisles—and the gendarmes got upset about the overflow of people anxious to get into the talk. Here is Wayne addressing the group on the subject of microcomputers and their growth in the U.S. . . . and the desperate need for software.

Wayne, on left, and M. David, in center, translating into French. The gist of the talk was that ways must be devised for software to keep pace with the hardware development in the microcomputer field if we are not to see a customer disillusionment that could result in a severe setback for this rapidly rising industry.

gramming language than we've been using.

After getting Adamed, we went to visit the new Byte Shop HQ. They have some exciting plans for setting up a zillion Byte Shops. Under the new ownership, I think they will be better financed and managed.

A lot of growth is being planned for in our industry . . . Computerland has a lot more stores in the works . . . and Tandy Computers is working on a network to cover the entire country. I'll be surprised if we don't have over 10,000 stores by August 1979 . . . counting the Radio Shacks, which are handling TRS-80s.

We rested for a few minutes and then headed for Oakland and dinner with Bill Godbout, George Morrow and Bob Mullen . . . flying out to a restaurant called New West in Godbout Air Force 1. The food was fantastic, but it was difficult to even notice this because we were having so much fun talking. I wish we could videotape a discussion among these three chaps. They are so full of ideas that 100 companies could spin off from a dinner like this.

The next morning Sherry and I were off to Los Angeles, then on to Atlanta as part of our Eastern Airlines special fare trip. For about $50 less than the round trip to Dallas we were able to get to Atlanta for three days, then L.A., then back to Atlanta (we stayed just one night at the Omni . . . what a great place that is!) . . . then on to Puerto Rico (we took a taxi downtown to sightsee El Moro, the fort defending the harbor entrance, and then back for our continuing flight) . . . and Martinique. From there we went back to Dallas for the hamfest/computerfest there in mid-June.

The Dallas ham/com didn't do badly for a first show. I gave a talk at 9 AM on Saturday on software, and the room was packed solid. I didn't expect more than a dozen people at that early hour, so I was amazed. Obviously, there were a lot of computer hobbyists turning out for this first microcomputer show in Dallas. This bodes well for the big computer show set for September in Dallas. I understand that just about all of the major firms will be exhibiting at that one.

I passed along my observation of the NCC show to the Tanners, who are running the Dallas show, and they decided to make an extra effort to make this show particularly attractive to computer store people, with several special sessions on financing, selling, advertising, marketing, etc. They might be able to work this into the biggest show of the year if they keep up such creative planning. I'll be on the program for a couple of talks.

Gold in Them 80s

For some reason Radio Shack seems to be playing it cool on the number of the TRS-80 systems they've been selling. Having personally seen most of the microcomputer factories, I strongly suspect that there are already more TRS-80 systems out there than everything else combined, even going right back to Altair serial-number 1. They are selling a mountain of them.

Oddly enough, though the other microcomputer manufacturers have worried themselves sick over how this might cut into their sales, so far the impact has been as I predicted: more business for everyone. If anyone is hurting for sales it is because their advertising and promotions are bad, not because people are not buying, and there are several firms with painfully poor advertising, sorry to say.

Getting back to the 80 and the possibilities for getting rich with this system: the possibilities are almost endless, if you put your imagination to work on it. Radio Shack is doing a reasonably good job of announcing new gadgets to work with their system, but their facilities and resources are limited, so the fact is that smaller firms will be able to run circles around them.

There are two ways for things to go . . . and I suspect that the second way will predominate. One is for many (or most) TRS-80 owners to buy an S-100 bus transfer unit, either from Radio Shack or some other vendor (I had my hot hands on the official Radio Shack transfer unit the other day, so it does exist, though still in prototype form). Then the S-100 boards for various applications can be used . . . for music, creative art, control of the world, telephone interface, etc.

The other way makes more sense, sorry to report, and that is for firms to come out with gadgets designed to plug into the 80 directly, using its bus. I saw my first such unit at Atlanta where one firm had a little box that plugged into the back of the 80 and produced Morse code when the letters were typed on the keyboard. Interface units that turn out to be popular will eventually be manufactured by Radio Shack, but by the time they lumber into production a small firm will be able to be in and out with perhaps a million or two in sales. Big firms just can't get into production as fast as small ones.

How many TRS-80s are there? Radio Shack admits to over 20,000 . . . and I would put it more in the 50,000 range, with perhaps the present production running close to 10,000 per month. Even with this large number being shipped there is a serious back-order situation. This is to be expected even though Radio Shack hasn't the money problems virtually every other firm in the business has.

With that size of a market out there, the need for accessories is obvious. What are you waiting for?

More Help Needed

Actually we need a whole lot more help . . . with *Kilobaud* and with the software development. Both appear to have excellent futures.

For *Kilobaud* we have a serious need for a technical editor. This calls for a combination background in microcomputers of hardware, software and systems. We've had this need for some time, and now, with John Craig moving on, the need is even more critical (that translates into more pay). There really isn't any way to run something like that remotely; we tried that and it flat didn't work acceptably for either us or the editor.

Each day at *Kilobaud* brings decisions that have to be made. We can tackle these in a quick (ha!) meeting when everyone is at hand; otherwise the decisions have to be made anyway, only without input from those not present. This inevitably results in paranoia setting in . . . often on both sides of the situation.

We're not sure how many people we'll be needing for our software production, but it will be more than a few. We'll be needing editors with a good knowledge of BASIC who can help evaluate programs submitted for publication. We'll need quality-control people, production people, etc. We'll need writers to prepare the documentation, to write advertising and to prepare catalogs of the programs.

(continued on page 21)

kilobaud™

Publisher/Editor
Wayne Green
Executive Vice President
Sherry Smythe
Assistant Publisher/Editor
Jeffrey D. DeTray
Managing Editor
John Barry
Editorial Assistants
Dennis Brisson
Susan Gross
Administrative Assistant
Dotty Gibson
Assistant Business Mgr.
Robert LaPointe
Production Department
Manager:
Lynn Panciera-Fraser
Staff:
Craig Brown
Gayle Cabana
Robert Drew
Bruce Hedin
Carl Jackson
James Lamothe
Dion Owens
Noel R. Self
Robin M. Sloan
Typesetting
Barbara J. Latti
Jennifer Johansson
Marie Walz
Photography
Bill Heydolph
Tedd Cluff
Associate Editors
Tim Barry
Bob Buckman
Ronal Burris
Sheila Clarke
Rich Didday
Phil Feldman
Doug Hogg, PhD
Tom Rugg
Peter Stark
Bookkeeper
Knud E. M. Keller
Marketing
Daniel D. Savage
Brian Coleman
Judy Waterman
Circulation
Pauline Johnstone
Customer Service
Joyce Tarr
Receptionist
Doni-Anne Jarvis
Computer Data Control
Judy Brumaghim
Linda Cate
Mary Kinzel
Computer Programming
Ron Cooke
Richard Dykema
Steven Lionel
Micro-Systems Lab
Edward C. Dow, Jr.
Mailroom
Theresa Toussaint
Bill Barry
Sue Chandler
Ethan Perry
Advertising
Heidi Kulish
Nancy Ciampa
Marcia Stone
Bill York
European Marketing Dir.
Reinhard Nedela
Australian Distributor
Katherine Thirkell
UK Distributor
L P Enterprises

Kilobaud is published monthly by 1001001, Inc., Peterborough NH 03458. Subscription rates in the U.S. and Canada are $18 for one year and $36 for three years. In Europe: Kilobaud erscheint montatlich bei Fachzeitschriftenvertrieb Monika Nedela, 7778 Markdorf, Markstr. 3. Abonnement DM 70.- plus Porto 7.20 + gesetzl. MWST; Sfr. 81.- plus Porto 7.20. Australia: For subscriptions write—Katherine Thirkell, Sontron Instruments, 17 Arawatta St., Carnegie, Vic. 3163 Australia. UK: £20 sterling/year—LP Enterprises, 313 Kingston Rd., Ilford, Essex, IG1 1PJ, England. Please write for other foreign rates. Second-class postage paid at Peterborough NH 03458 and at additional mailing offices. Publication No. 346690. Phone: 603-924-3873. INCLUDE OLD ADDRESS AND ZIP CODE WITH ADDRESS CHANGE NOTIFICATION.

BOOKS

BASIC and the Personal Computer
Thomas Dwyer, Margot Critchfield
Addison-Wesley, Reading MA
440 pages, $12.95

I can say, without reservation, that this is the best BASIC book I have ever seen. Explanations, examples, and applications fill every page. This is certainly not a dry book, and both the novice and not-so-novice will gain much from it. Critchfield and Dwyer make it clear that they fully understand the mechanics of learning. I believe that this would make an excellent text for a BASIC programming class. However, for the book to provide the most benefit, the individual or class should have access to a BASIC computer.

Most of the programs are written in Maxi-BASIC. But just in case your BASIC interpreter does not support all of the features that the authors' does, many variations are provided to allow you to make them fit just about any version. In fact, converting a program to your particular BASIC is also a very good way to get to know it better.

The book starts with an overview of personal computing equipment and programming and then moves into an explanation of the BASIC language. This instruction takes the form of eight hourly sessions, which in turn are broken down into numerous exercises. The practice programs are always interesting.

It is tempting to move too quickly through this area, especially if you already have some BASIC background. You must avoid this temptation as there is much to learn, and each example should be examined and tried. These include computer-assisted educational math, batting averages, craps, balancing a checkbook and many more.

Chapter three goes into computer graphics. A variety of excellent examples for plotting and scaling is given. Of interest to many will be the Weight Watchers Record, which keeps track of weekly weight gain or loss (you are losing weight aren't you?) and then plots a graph of the results. Saving programs on tape and disk is also covered.

Word processing is the theme of chapter four. String handling and manipulation get special attention. Would you like your computer to write poetry or print your letters for you? How about word games? These are all to be found here.

Chapter five involves matrices and sorting routines. Many different sorting schemes are discussed along with their advantages and disadvantages. Bubble, selection, shell and quick sorts are explained and programmed for you. If you want to sort strings instead of numbers, you can do that also.

Also included in this chapter is the explanation and writing of a football scout's record-keeping and report-writing program. I don't know if we have any scouts in our midst, but the very same programming techniques can be used in many other fields. The generated report relates data in an easily understood manner and displays trends that can be more important than the data itself.

Chapter six concerns computer games. While many games are educational as well as entertaining, whenever you mention computer games you get many different reactions. To some hobbyists games are the beginning and the end of personal computing, while others have no use for them at all. The frequently overlooked fact is that writing or modifying game programs is a very good method of improving your programming skills.

Do you want to write your own games for fun (or profit)? The necessary techniques are well covered. Want to play spies on a grid, poker, crazy eights, or horse race? How about archery or Planet-X landing? These games are all discussed, programmed and run.

Graphs and computer art are the theme of chapter seven. Now we are getting into more complex algorithms. It took my SOL over two hours to print one page of the electric field intensity display program. The plotting possibilities with a standard alphanumeric terminal are impressive. All kinds of charts and graphs can be generated . . . plot multiple functions, bar graphs or just draw pictures. It can all be done with BASIC if you know how.

When data bases and files are mentioned, many hobbyists dismiss them as business-related items. Nothing could be further from the truth. Most programs have some kind of data base, and for storing and retrieving information, files can't be beat. Three comprehensive programs are detailed in chapter eight; they show the difference between storing data within the program itself and storing it externally in a cassette tape file. Diet information, food data and charge-account records are the examples used.

Chapter nine covers computer simulations. Simulate means to copy, and computer simulations are programs that imitate a real function. For instance, the Planet-X lander game is a primitive simulation of a real landing on the moon. In fact, almost any computer program is simulating something, whether it be dealing cards, playing chess or predicting future business trends. In this chapter there are two major examples; one simulates the interaction between the number of customers and the employee requirements of a retail store and the other the operation of a pseudo-government on a distant planet.

Finally, chapter ten looks ahead to more advanced subjects. Color graphics, light pens, music and analog I/O are some of the items discussed. Four pages of beautiful color plates show the Cromemco Dazzler and the Compucolor units in action. There can be no doubt that all displays will be in color someday.

Given a reasonable amount of attention, this book can't help but benefit the reader. Self tests are scattered throughout. The examples are always easy to understand because of the authors' clear explanations. Anyone planning to write a computer textbook should study the instruction methods used in this book.

Two outstanding features are the Project ideas and the Style corner. Project ideas are suggestions for modifying programs beyond what has been presented in the text. These would make excellent homework assignments for a BASIC classroom situation. The style corner is written by John Nevison and shows some of the programs written using the rules in his book, *The Little Book of BASIC Style,* which is also available from Addison-Wesley Publishing Company.

Authors Critchfield and Dwyer hint that a more advanced volume might follow this one. I certainly hope so because I am thoroughly sold on *BASIC and the Personal Computer,* and I am looking forward to more of the same.

Rod Hallen
Tombstone AZ

Z-80 Programming for Logic Design
Adam Osborne *et al.*
Osborne & Associates, Inc.
Berkeley CA, $8.50

This book is the third in a series from Adam Osborne that includes similar books for the 8080 and 6800. All three books are designed as follow-ups to *An Introduction to Microcomputers* from the same author. This book shows you how to write assembly-language programs for your Z-80 that simulate the actions of flip-flops, gates, timers and other logic elements. Using this technique, you can simulate circuits containing many integrated circuits, thereby reducing the hardware required for any given application.

At the outset, the author demonstrates how to simulate the functions of individual gates and flip-flops, using relatively short assembly-language routines. Virtually any logic function can be implemented with only a handful of assembly-language statements.

The next section of the book puts this basic information to a practical use. The techniques of logic simulation are used to recreate the circuitry needed to control the print wheel of a Qume Q-series or Sprint Series printer. This is a daisy-wheel type printer, and the portion of the printer interface to be simulated contains six flip-flops, three one-shots, a 555 timer and about two dozen gates and inverters. We're taken step by step through the simulation, during which we are shown how each of the logic elements can be simulated. At the conclusion of the chapter, we have written a 140 line assembly-language

(continued on page 21)

THERE'S A NEW ADDITION TO THE HAZELTINE FAMILY...AND

SYNCHRO-SOUND'S GOT IT!

THE ALL-NEW
HIGH-PERFORMANCE LOW COST

HAZELTINE 1400 VIDEO TERMINAL

All 128 ASCII Codes • 64 Displayable Characters • High Resolution 5 X 7 Dot Matrix • Cursor Addressing and Sensing EIA Interface • Rates up to 9600 Baud • Self Test

$27.51 per month*

$799.00 complete

HAZELTINE 1500 VIDEO TERMINAL

- Reverse video
- 24 X 80 display
- Programmable brightness levels
- RS232 and current loop . . . and much more

$39.57 per month*

$1149.00 complete

995.00 kit

HAZELTINE MODULAR 1 EDIT INTELLIGENT TERMINAL

- 1920 character display
- 8 different video levels
- Full editing capability
- Removable keyboard . . . and much more

$57.14 per month*

$1659.00 complete

*36 month lease
Maintenance additional
OEM pricing available

SYNCHRO-SOUND ENTERPRISES, INC.

The Computer People
193-25 Jamaica Avenue, Jamaica, New York 11423 Dept. HK
212/468-7067 TWX 710-582-5886
West Coast: 5810 Commerce Blvd.
Rohnert Park, CA. 94928 • 707/544-2865

Hours 9-4 Daily and Saturday

Visit our new showroom
Working units on display

BankAmericard Master Charge

TROUBLE-SHOOTERS' CORNER

Ralph Wells

In an article titled "Cooperation" in the June 1978 *Interface Age,* John Craig (former editor of *Kilobaud*) quoted Bob Jones (publisher of *Interface Age*) as saying, "If we all work together at this thing, then we're all going to be successful together." Last month in this column, I tried to emphasize the importance of just this type of cooperation in the field of troubleshooting and used the phrase *synergistic synectics* to describe it.

At about the same time I read John's quote, I also received my current (May 20) copy of *EDN* magazine, containing an article by John Conway entitled "Glitches Can Turn Your 'Simple' Interface Task Into a Nightmare." For anyone interested in troubleshooting microcomputers, this *EDN* article should be required reading. It recounts the debugging of a glitch in the Apple II and reads like a Sherlock Holmes mystery. It's well illustrated and fascinating reading, but at the end I felt that I'd been on a witch-hunt. There was a haunting feeling that the *EDN* group might prefer to see the Apple II, its mentor, Steve Wozniak and *all* 6502 micros in the stocks on a town square in New England.

Glitches—Part II

I'd been contemplating designing in an Apple II as part of a system development for Inmarco (where I work). The requirements called for it to be interfaced to read and write through the 6522 VIA (versatile interface adapter). The gist of the *EDN* article was that the difficulties in using an Apple II for this purpose were so serious that *EDN* was considering discontinuing one of the best series of *practical* development (Apple-Indecomp project) articles I've had the pleasure of following.

After I mulled it over for a couple of days, I bought an Apple. It arrived last Wednesday afternoon. By Friday night we had it working through a kludged VIA. What happened in between is very much like the second episode of a multipart serial TV drama, and just the sort of situation I'd been looking for to illustrate some basic troubleshooting principles.

When last we left this stirring tale . . . Some background is in order. The *EDN* staff developed and beautifully documented an I/O port using a 6820 PIA (peripheral interface adapter). It worked on a 6800 system and on a KIM, but not on Apple II. After some hair-raising adventures (including a fall down an elevator shaft by one of the staff), the villain was tracked down.

In the last scene, our heroes are provided with a potential solution to the basic problem. In the closing scene the *EDN* staff is left with several multiple-choice imponderables: Should they (a) redesign their "simple" interface to include a delay-producing (expensive) buffer, (b) give up on everything or (c) something in between?

The villain turned out to be a Ø0 (phase zero) clock line masquerading as Ø2, aided and abetted by the $\bar{Ø}0$ (inverted Ø0) line passing itself off as Ø1. The *real* Ø2 never even got to the user bus as proclaimed.

As we resume our story, we find . . . The scene shifts to the Sunset Strip in glamorous Hollywood, where we pick up the action as an Apple II is being delivered to the Inmarco engineering department by a couple of Apple lovers. These gentlemen are convinced that the *EDN* article is a gross miscarriage of justice and that, even if there *is* something amiss, it can be easily remedied—not to worry. Billy Shatto (my partner in this adventure) had kludged a VIA using Vector Slit 'N Wrap so that we could easily change the design, and change it we did.

The well-documented *EDN* article provided some schematics and clues. Also helpful was that Apple was preceded by three earlier 6502 micros in my personal collection: Jolt, KIM and PET.

An Inventory of "Tools"

Inmarco has a resident Apple freak named Dave Gordon. If you tear away his accounting-executive facade, you'll find the most turned-on Apple polisher that ever lived. If you read last month's column, you can recall that I feel one of the first steps in practical troubleshooting is to inventory your *tools*, or assets. I also had a lot to say about the "most valuable tool" being synergistic, synectic friendship of the type I referred to in the beginning of this month's column.

Dave is another excellent example of what I'm referring to. He'd be the first to admit that, technically, as an engineer or programmer, he's a great accountant, and this was certainly a technical troubleshooting problem. Even so, I'd rate Dave as the top asset we had at our disposal. We also had experience with interfacing I/O to 6502 microprocessors, since both the Jolt and KIM used PIAs, and the PET used VIAs. We had also interfaced an SWTP printer (PR-40) to both the PET and Dave's Apple, and then used his Apple to write the machine-language software to drive the PET's VIA (through the user plug) from BASIC. The most valuable experience had been with the Apple-printer interface because it used a 6520 PIA and 2708 PROM, both addressed with the bogus Ø2.

For the most part the printer worked, but the uneasy aspect of this experience was that it did *not* do all the things we expected it to do—more on that later. We didn't have the Hewlett-Packard #1615A logic analyzer ($6800) that *EDN* used, but we did have a four-channel Tektronics #549 storage scope and a Paratronics analyzer (which we didn't have to use). We also had some logic probes, DVMs, etc. This inventory of assets seemed to warrant taking the risk, so here we were with an Apple.

And We're Off . . .

After unpacking the Apple II, we ran through a few routine checks with Dave's help. We decided we'd go for broke and try the VIA configuration as shown in Fig. 2. I was pretty sure that the problem could be solved by delaying the Ø0 (used as Ø2 in the Apple). Putting Ø0 through two stages of CMOS should delay it at least as much as the delay caused by the CPU (see Fig. 1) and gate

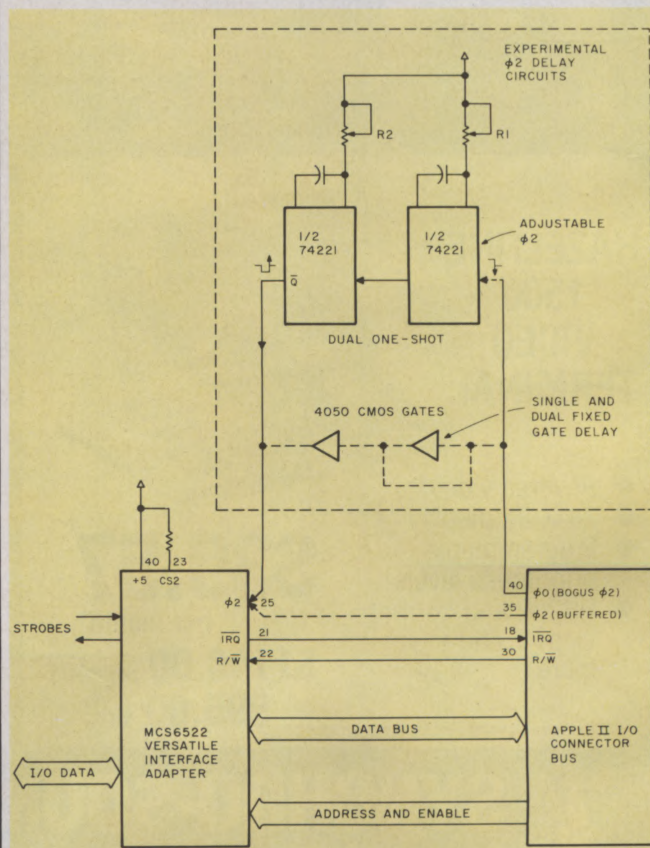


Fig. 1. Experimental versatile interface adapter for Apple II using various methods to provide delay for Ø2 input to pin 25 on the 6522. A buffered Ø2 from the CPU was distributed on unused bus #25 in the last analysis.

B11 (the latter wasn't mentioned in the *EDN* article).

Billy wrote a BASIC program to POKE in the VIA initialization data followed by a WRITE loop. Then he fixed it up. Nothing came out. Was *EDN* right? Had we goofed? (We had.)

Failures like this are a daily occurrence for Billy and me, but Dave was suffering real pain . . . something akin to an expectant father at delivery time. We juggled the software around and looked at some waveforms. The double CMOS gate delay was a lot longer than I had expected, so we cut it down to one—still no output. Then we substituted the PR-40 printer interface. It worked with its own PROM program, but when Billy tried it with the BASIC driver (suitably modified), nothing came out and the day was over.

We changed to the variable design of Fig. 2 so that we could adjust both the leading and trailing edges of an "adjustable" Ø2 coming out of the 74122, and Billy rewired it Thursday morning. By the time I arrived at work, he had discovered that the PIA could be initialized and driven by a machine-language program or from the Apple monitor, but *not* from a POKE command in BASIC. The scope showed two access pulses where we expected to see one. The first was coincident with the proper data pulse—the second wasn't. It was turning itself off about as fast as it was turning on!

We felt that the major problem was still the hardware, and it would be driven by machine language anyway, so Billy rewrote the test program in machine code, and now the PIA worked! The *EDN* article quoted Steve Wozniak (the chief Apple grower) as saying that the 6502 had an odd habit of generating a READ before a WRITE during store operation. My experience with the Jolt and PET didn't jibe with that. I don't like loose ends like this floating around, but for the time being we decided to ignore BASIC and stick with machine code. (Days later, we found a bug in our program.)

Billy changed the program back to double-delayed Ø0 VIA, and we tried to write with it. Success! We could move the critical edges back and forth by tweaking R1 and R2. When the rising edge of this "adjustable" Ø2 was aligned with the Ø0 (pin 40), the VIA would fail. When we moved it to coincide with the real Ø2 on pin 39 of the 6502 or as much as 200 ns beyond that time, it worked! So much for "writing" out with the VIA.

Wozniak was also quoted as saying that the PIA was probably only useful as a write-out-to device. Our project called for read-after-write on a 9-track tape deck at 25 ips. That meant that we would have to both *read* and write, synchronize and index in less than 40 usec—could we do it with the Apple?

Our 6800 had been too slow. Billy rewrote the program to read and display in a high-speed loop. It worked! I was relieved, Billy was happy and Dave was ecstatic. His faith in his beloved Apple was vindicated. When we told him that it wasn't over yet and that we had *no* intention of putting adjustable pulses on every I/O board with a PIA or VIA, it cooled him off a little, but he walked off into the sunset muttering something about "that being only a *little* problem."

A Can of Green Worms

In last month's column I said that the second step in troubleshooting was to define the problem and that the third step was to "fix" it. By now we felt we had the fundamental problem fairly well defined and turned our attention toward what to do about it. We knew that the real Ø2 coming from pin 39 on the 6502 was the correct phase and formed to do what the chip manufacturers (MOS Technology, Synertek and Rockwell) said it should do, so why not just use it instead of the Ø0?

Fourteen MHz works out to have around a 35 ns pulse width, and the Ø2 phase error was almost that much. Would replacing the Ø0 with a Ø2 for the whole system be the answer? If the feedback circuit didn't cause a race condition, we might just get away with it. We decided to try.

Billy cut the etch so that only the gate (B11) was driven from B1 (74S175 D flip-flop wired to toggle). The real Ø2 from pin 39 on the CPU was buffered by a TTL chip we installed on the kludge section of the Apple (a premeditated plus in the Apple design).

The result was a disaster. Attempts to trace the problem through the feedback loops produced a jumble of unintelligible green squiggles on the four-channel scope. We had opened a can of worms—green ones. We quickly clapped the lid back on and restored the original Ø0 wiring . . . and sought a practical compromise.

Our past experience with memories, both static and dynamic, indicated that the phase error probably wouldn't bother them either way; so when it came right down to it, the only critical devices were most likely those on the interface bus. An examination showed that there were two unused lines (three if you count the user). One of them (pin 35) was adjacent to the other clock pulses, so Billy reconnected the original drive circuit and then ran a line from the real Ø2 down through the buffer (see Fig. 1) and back to pin 35 on the interface bus.

Now when we tried our VIA kludge (without a delay on board), it worked when driven from the "new Ø2" and failed when driven from the original line. This means that if we use VIAs, PIAs or the unbuffered PR-40 driver in our own designs, we will use the "new" *real* Ø2 line on pin 35. If we use a buffered design, such as the Apple PROM burner described in *EDN*, we can have a choice. Either line *should* work, but to play it safe I'd use the real (buffered) Ø2.

Later we tried the Ø2 right out of the CPU (no buffer or buffer delay). If we routed the wire *exactly* right (and the moon was full), then it would work, but it needed the extra delay as a safety margin.

Will the Real Ø1 Please Stand Up!

So far we've been discussing the use, misuse and misnomering of Ø0 and Ø2 (see Fig. 3). What about Ø1? We did a lot of worrying about Ø1, probably unnecessarily. To start with, the waveform generated by our Apple 6502 Ø1 doesn't match the specifications shown in the MOS Technology book or the *EDN* article. Ø1 is supposed to be shorter than Ø2 on both ends. The leading edge was OK, but the trailing edge was coincident (as best we could read it on our scope), or maybe a few nanoseconds longer than Ø2.

On Friday afternoon after we got Ø2 straightened out, we checked the KIM and our just-arrived VIM 1, which used a Synertek 6502. The KIM looked just like the Apple, but the VIM 1 looked as the textbook said it should. With the exeception of a clamp diode, the crystal clock circuits of the KIM and VIM are

(continued on page 116)

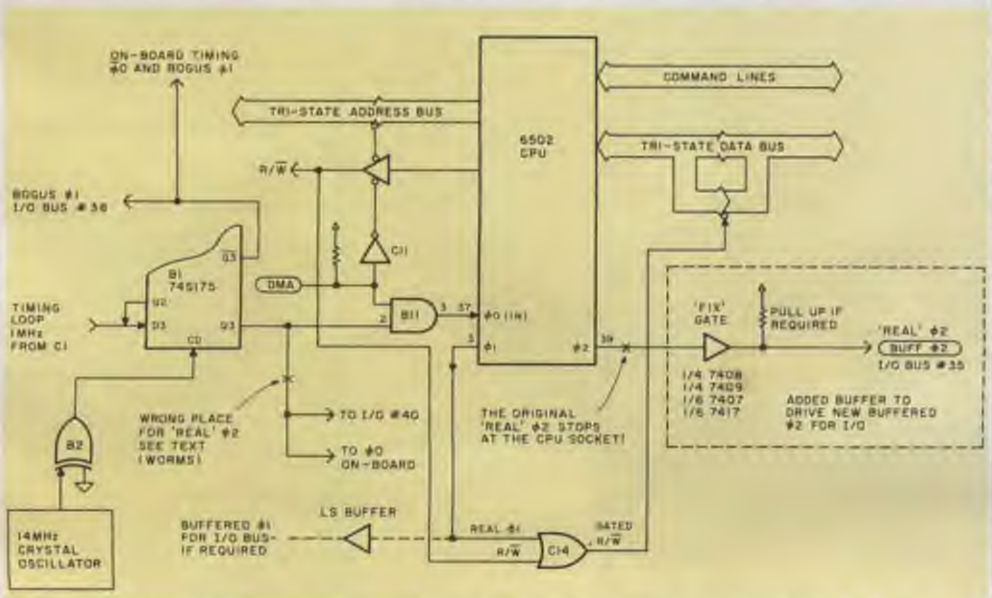


Fig. 2. Apple II clock circuitry showing "fix" buffer and waveform generation described in text.

NEW PRODUCTS

Software Package for COSMAC Micromonitor

RCA's Micromonitor Operating System (MOPS) CDP18S831 provides Micromonitor users with enhanced debugging techniques ranging from simple terminal Micromonitor dialog to hands-off system testing with commands coming from disk files. The MOPS software supplements the performance of the RCA COSMAC Micromonitor CDP18S030 by providing the user access to the processing and storage capabilities of the COSMAC Development System (CDS) II CDP18S005 equipped with the Floppy Disk System CDP18S805. The Micromonitor, a self-contained debugging tool for use with any system based on the CDP1802 COSMAC microprocessor, permits in-circuit debugging in real time so that both hardware and software problems can be efficiently identified.

The package consists of a MOPS diskette plus a UART module and connecting cable to interface the Micromonitor to the CDS. MOPS provides an extended set of Micromonitor-type commands: (1) to conveniently switch Micromonitor commands and responses to and from a variety of system peripherals; (2) to more completely interrogate the CPU state; (3) for loading the system-under-test from a disk file; (4) for saving the system-under-test memory, registers, etc., in disk file; (5) to facilitate automation in system debugging and testing.

In single quantities, MOPS is priced at $350 and the Micromonitor at $1600 (both U.S. only).

RCA Solid State Division, Box 3200, Somerville NJ 08876.

RCA's Micromonitor Operating System.

Manual for Apple II

Apple Computer, Inc., announces the availability of an easy-to-read, illustrated manual for its Apple II computer system. Entitled *Apple II Basic Programming Manual*, the book was authored by Jef Raskin, a computer professional who has written and lectured extensively on the subject of computer science to both the novice and the professional.

The book assumes no prior reader background in programming or computers. Programming is explained in everyday English with no computer jargon used. Moreover, the book introduces the whole computer to the reader. Thus, unlike programming manuals that solely teach a language, this book teaches a language in the context of the computer in which it will be executed.

Four chapters comprise the manual. Chapter 1 guides the reader through the details involved in connecting the various Apple II system elements, television, tape cassette player, etc., and describes the computer's control functions. The second chapter starts the reader programming with the BASIC Programming Language using simple, colorful examples. Chapter 3 moves the reader into writing complete BASIC programs by providing detailed information on most BASIC language commands. Finally, the last chapter describes strings, arrays and subroutines for the reader who has acquired an understanding of the BASIC language and is ready to write more extensive programs.

The manual is presently available from Apple dealers for $5.95.

Apple Computer, Inc., 10260 Bandley Dr., Cupertino CA 95014.

Apple II programming manual.

Tandy 10 Business System

The Tandy 10 system.

The Tandy 10 system consists of a work station with diskette drives integrated into a compact metal desk and a separate matrix printer that prints 60 characters per second. Faster printers are available as options. The work station includes a video display, professional standard typewriter keyboard, 10-key calculator pad for numeric entry and 15 special function keys for data editing. With optional peripherals, it can be used as a terminal to access larger data systems.

Each diskette can hold up to 256,000 characters, providing a total of more than ½ million characters on line. Internal memory capacity is 40,960 characters. Screen formatting language allows user prompting for data input. The Tandy 10 comes with extended BASIC. Fortran IV and Assembly Level program languages are also available as options.

The complete business computer system with work station, diskette drives and matrix printer is priced at $9950. Tandy Computers, Department R22, PO Box 2932, Fort Worth TX 76101.

Software from ASI

Administrative Systems, Inc., 222 Milwaukee, Suite 102, Denver CO 80206, announces their latest software package for 8080 and Z-80 microcomputers. The S.O.S. (Single-user Operating System) package provides the user with a step between the OPUS stand-alone high-level languages and the TEMPOS Multi-User/Multi-Tasking Operating System.

Incorporating many of the features of TEMPOS, the S.O.S. package includes:

OPUS/THREE—the high-level Compiler/Interpreter from ASI;
TEXTED—an easy-to-use, line-oriented text editor;
ASSEMBL—an 8080 assembler;
FILES—a diskette file manipulator;
UTILITIES 1—a package of 12 utilities programs.

Full upward compatibility has been retained to allow the user of S.O.S. to access data and programs developed at lower levels; all may be used under the TEMPOS Operating System as well. All floppy disk and serial device I/O is handled by S.O.S.; a System Generation routine lets the

Centronics' acoustic package installed .

Automated Systems' Power Master system.

user define I/O drivers as required, or select from the standard drivers provided by ASI.

The recommended hardware configuration includes an 8080 or Z-80 processor, 32K of RAM memory, one or two floppy-disk drives and terminals as required. The system typically resides in less than 10K of RAM. The package is priced at $385; the User's Manual Set may be purchased separately for $20, which may be credited toward purchase of the S.O.S. package.

Acoustics Package for Centronics Printers

In response to the demand for a quiet, efficient line printer that will not interfere with the day-to-day operations within the office, Centronics Data Computer Corp. has introduced an optional acoustic package for use on all four members of its 6000 Series of high-quality, fully formed character line printers.

The factory-installed option consists of several distinct elements, each of which is designed to ensure quiet operation of the printer. The units, which will carry an end user price of $450, are currently in production. Centronics, Hudson NH 03051.

SUPER-MONITOR for Poly

A new SUPER-MONITOR for PolyMorphic System 88 owners having trouble running software orged at 0000H due to the onboard system monitor is now available. It permits the use of low off-board RAM for program storage. The SUPER-MONITOR plugs into the two remaining ROM sockets on the Poly processor card and leaves the original monitor intact. The extended monitor contains many useful features that will give enhanced performance to the Poly 88, as well as free up low memory.

The SUPER-MONITOR features: DUMP, MOVE, VERIFY, EPROM PROGRAMMER, FILL, IN, OUT, CASSETTE SAVE, CASSETTE LOAD, SEARCH, SERIAL PORT DRIVER, GOTO, MEMORY MODIFY and RETURN TO POLY MONITOR. It is available as a two-chip set on 2708 EPROMs with complete documentation for only $59, plus shipping.

Computer Hobbies Unlimited, 9215 Midlothian Turnpike, Richmond VA 23235.

Transient Voltage Suppressor Protects Circuitry

The HDA Power Master transient voltage suppressor dissipates destructive electrical transients produced by changing loads, switching SCR drive systems and operation of most electrical equipment.

Transients are a system problem resulting from variations fed back into the electrical network from operating equipment—causing vibration, noise, excessive wear, heating in electric equipment and the waste of electrical energy. This system can significantly reduce these transients and protect delicate computer circuits and electronic systems.

Automated Systems, Inc., 5265 Port Royal Rd., Springfield VA 22151.

Graphic/Text Video Interface

MiniTerm Associates, Inc., Dundee Park, Andover MA 01810, has developed MERLIN, a combination text and graphic video display board that combines functions of text display, graphic display (320 H by 200 V resolution), keyboard input port and 4K bytes of on-board control ROM. Assembled and tested, the MERLIN video interface with 4K control firmware and super-dense graphic option is priced under $500. MERLIN is also available in kit form without ROM software for less than $300.

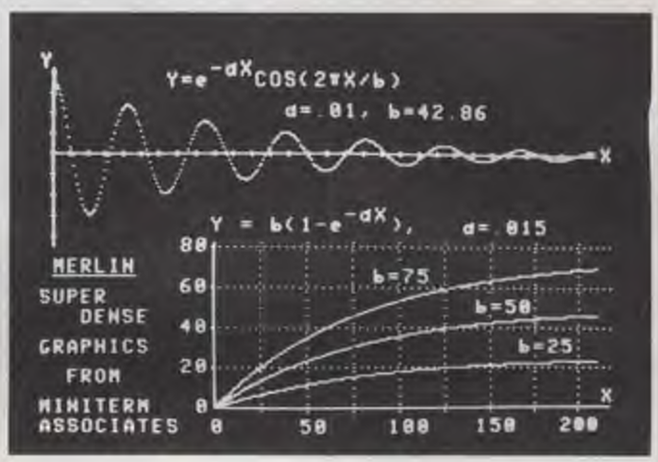
The MERLIN video interface, plus keyboard and video monitor, is a console I/O device for the small system (requires only mainframe, CPU and memory for a "total" system). MERLIN displays 20 lines of easily readable text with 40 characters per line. This is suitable for text editing, BASIC and assembly programs and large screen classroom use. Both upper and lowercase characters can be displayed. MERLIN is also a medium-resolution graphic display for graphic development and end-user systems. The standard resolution is 160 H by 100 V, true bit-mapped graphics. This can be increased to 320 H by 200 V (64,000 bits per screen) with the super-dense option.

The 4K control ROM firmware provides a keyboard driver with special "Edit key" decoding, display output with auto and "wrap-around" scrolling, 14 monitor functions, 25 cursor/edit functions plus graphic subroutines and a keyboard graphic drawing mode. The firmware can be tailored to your individual system with four user-definable monitor keys, seven edit keys and selectable I/O drivers.

The MERLIN video interface provides the main console I/O in a small system, or can be the heart of a sophisticated graphic development system. MERLIN is also economical and versatile enough for OEM graphic applications. In any application where there is a need for real time plotting, complex equation plotting, fine line drawing or pattern placement and/or movement (e.g., games), MERLIN can fill the job.

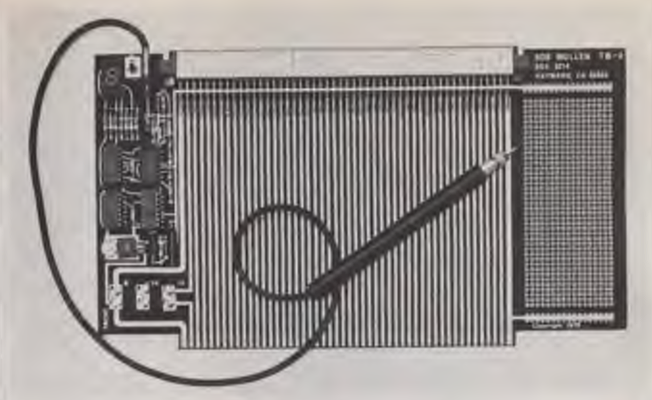
Word Processor for the PET

CONNECTICUT microCOMPUTER announces a word pro-



MERLIN's super-dense graphics display.

88/MS mass storage unit.



The Mullen TB-2 board.

cessor program for the Commodore PET. This program permits composing and printing letters, flyers, advertisements, manuscripts, articles, etc., using the Commodore PET and an RS-232 printer.

Script directives include line length, left margin, centering and skip. Edit commands allow the user to insert lines, delete lines, move lines, change strings, save onto cassette, load from cassette, move up, move down, print and type. The Word Processor Program addresses an RS-232 printer through a CmC printer adapter. The program costs $29.50 postpaid.

CONNECTICUT microCOMPUTER, 150 Pocono Rd., Brookfield CT 06804.

Real Estate Management System

The MANAGER is a general-purpose real estate property management package that will keep track of 500 tenants per diskette. It will store for immediate recall the following items on each tenant: rate of rent on unit; name and address of tenant; date and amount of last payment; current status of tenant (e.g., current, late, overdue, three-day notice); amount of last month rent held; amount of security deposit held.

The package has a daily utility program that generates tenant billing and overdue notices. The tenant data-base is automatically updated each day. A complete record is kept of funds in the owner's trust account. There are provisions for a year-to-date or month-to-date owner's trust-account report.

Other programs included with The MANAGER will print out a profit/loss statement for the owner, a report for the resident manager, and a fast report for you to use when answering owner or tenant questions over the phone. The software is designed to run in 24K of memory using any combination of terminal and printer. The MANAGER comes complete on diskette in North Star BASIC and includes an extensive manual. $199 shipped postpaid. California residents, add 6 percent sales tax.

Alpha Data Systems, Box 267, Santa Barbara CA 93102.

PolyMorphic's 88/MS

PolyMorphic Systems' 88/MS increases the storage capabilities of System 88 microcomputers through the use of disks that are not only larger than minifloppy disks, but will also store twice as much information per square inch and store it on both sides. One disk can hold 1.2 Mb—more than 500 pages of text.

The 88/MS consists of two drives for 8-inch magnetic storage disks in a walnut cabinet with brushed aluminum front panel.

A System 88 microcomputer with one or two 88/MS units will handle all the files and processing needs of most small businesses and professional offices. Present owners of any System 88 microcomputer can add the 88/MS mass storage unit with no changes in their equipment's operating system. Ready-to-use packages for doing such tasks as accounts receivable are available.

PolyMorphic Systems, Inc., 460 Ward Drive, Santa Barbara CA 93111.

New TB-2 Extender Board

The Mullen TB-2 Extender Board Kit retains the price of its predecessor but offers several new features. The built-in logic probe now reads out into a 7-segment display and also includes a pulse catcher plus an LED whose brightness corresponds to the duty cycle of a pulse stream. A general purpose "kluge board" section (with holes on 0.1 inch grid) aids development of circuits used in debugging or testing; an on-board 5 V @ 1 A regulator powers this section as well as the logic probe.

The TB-2 also incorporates features of the previous model, such as links in the power supply lines for current measurement/fusing/independent supply switching; an edge connector label that identifies power, ground and S-100 bus signal pins; full-width board size to allow use of card guides; and gold-plated edge-connector teeth that stand up to repeated insertions.

The Mullen Extender Board Kit lists for $35 and is available at many computer stores or by direct mail from Mullen Computer Products, PO Box 6214, Hayward CA 94545.

Extended Monitor for KIM

"XIM" (Extended I/O Monitor) is a programming and debugging package for the KIM-1 microcomputer. It provides commands to move, compare or search blocks of data for strings, set breakpoints, calculate hexadecimal branch displacements, load and print ASCII text, dump hex data on a terminal in 16 by 16 matrix form from memory and display processor registers.

"XIM" requires 1K of core at 2000 hex and is easily relocated to suit the user. It is also ROMable and easily modified, and features 17 commands, including four user-defined commands for expansion. The package includes a 45-page manual with source and object listings and a KIM-compatible cassette tape for $12, postage paid USA and Canada.

Pyramid Data Systems, 6 Terrace Ave., New Egypt NJ 08533.

MicroPower Center For Energy Control

Energy Technology, Inc., developers of the Coby 1 System for remote control of electrical equipment, announces MicroPower Center 168, a microcomputer for energy management in small-commercial or light-indus-

MicroPower Center 168 by Energy Technology.

Look To The North Star HORIZON Computer.

HORIZON™—a complete, high-performance microprocessor system with integrated floppy disk memory. HORIZON is attractive, professionally engineered, and ideal for business, educational and personal applications.

To begin programming in extended BASIC, merely add a CRT or hard-copy terminal. HORIZON-1 includes a Z80A processor, 16K RAM, minifloppy™ disk and 12-slot S-100 motherboard with serial terminal interface — all standard equipment.

WHAT ABOUT PERFORMANCE?

The Z80A processor operates at 4MHZ — double the power of the 8080. And our 16K RAM board lets the Z80A *execute at full speed.* HORIZON can load or save a 10K byte disk program in less than 2 seconds. Each diskette can store 90K bytes.

AND SOFTWARE, TOO

HORIZON includes the North Star Disk Operating System and full extended BASIC on diskette ready at power-on. Our BASIC, now in widespread use, has everything desired in a BASIC, including sequential and random disk files, formatted output, a powerful line editor, strings, machine language CALL and more.

EXPAND YOUR HORIZON

Also available—Hardware floating point board (FPB); additional 16K memory boards with parity option. Add a second disk drive and you have HORIZON-2. Economical serial and parallel I/O ports may be installed on the motherboard. Many widely available S-100 bus peripheral boards can be added to HORIZON.

QUALITY AT THE RIGHT PRICE

HORIZON processor board, RAM, FPB and MICRO DISK SYSTEM can be bought separately for either Z80 or 8080 S-100 bus systems.

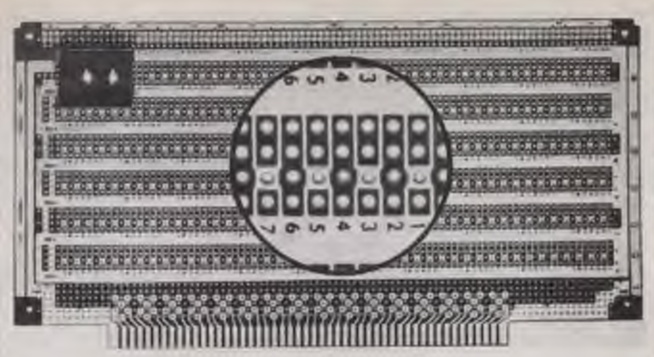
HORIZON-1 $1599 kit; $1899 assembled.
HORIZON-2 $1999 kit; $2349 assembled.

16K RAM—$399 kit; $459 assembled; Parity option $39 kit; $59 assembled. FPB $259 kit; $359 assembled. Z80 board $199 kit; $259 assembled. Prices subject to change. HORIZON offered in choice of wood or blue metal cover at no extra charge.

Write for free color catalogue or visit your local computer store.

NORTH STAR COMPUTERS N9
2547 Ninth Street • Berkeley, California 94710 • (415) 549-0858

OAE's PP-2716.



VE's ANY DIP Plugbord.

trial businesses.

MicroPower Center 168 (the 168 signifies the total number of hours in a week and its ability to set up 7-day cycles, or repeat patterns) can automatically turn the current on or off at designated locations around a business. This user-programmable electrical load controller can be used, for instance, to turn the lights or any other equipment off at a preset time each night or at different times on different nights.

The user can decide in advance when he wants any device enabled (supplied with power). The MPC 168 can handle on-off signals to as many as 100 locations around a small business. A 12-hour battery backup system will keep the Center functioning in the event of a temporary power outage.

The MicroPower Center uses the four varieties of Coby remotes to supply or withhold electrical energy from wall switches, wall plugs or to installed wiring (24-volt or 220-volt). The typical MicroPower Center 168 installation in a small business would run between $5000-6000. Deliveries begin in August 1978.

Energy Technology, Inc., 204 Conway (PO Box Q), Las Cruces NM 88001.

PP-2716 PROM Programmer

Oliver Advanced Engineering, Inc., announces the PP-2716 for the unique single-supply Intel 2716 EPROM. A 5 foot flat ribbon cable connects this full-feature programmer to *any* read only PROM socket via a 24-pin plug. With OAE's PROM socket interface, data is sent over the eight lower address lines to the programmer. No additional power supplies are required, and all timing and control sequences are handled by the programmer.

Each programmer comes complete with an internal DC-to-DC switching regulator and zero insertion force socket. The unit is packaged in a handsome anodized aluminum case for tabletop operation. Each unit comes completely assembled, tested, aligned and ready for use. Simply plug the unit into any read only socket and go! Price is $295.

Oliver Advanced Engineering, Inc., 676 West Wilson Ave., Glendale CA 91203.

* * *

Retail prices on the OP-80A Paper Tape Reader have increased from $74.50 to $84.50 kit, $99.95 assembled and tested.

Vector Electronic 8804 Board

Vector Electronic's Model 8804 board, bus and shape compatible with Altair and Imsai microcomputer boards, holds up to seventy 14- or 16-pin DIPs or any combination of DIP sockets with 0.3, 0.4, 0.6 or 0.9 inch lead spacing. Designed as the "ANY DIP" Plugbord, the 5.3 inch by 10 inch by 0.062 inch board has 100 (50 each side) card-edge contacts on 0.125 inch centers to accommodate an S-100 bus organization. Two 100-hole rows of individual 0.1 inch spaced pads across the top of the board permit additional input/output via ribbon-wire assemblies. Inexpensive wrap posts may be fabricated by inserting Vector's T46-5-9 wrapped-wire pins into the holes.

For wiring convenience, power and ground buses are in an offset ladder pattern on opposite board sides. One corner of the 8804 may be used for a low-profile heat sink with two regulators in the TO-220 packages. The leads of one regulator position are prewired to power input, ground and regulated power. The other regulator position is uncommitted. The circuit boards have complete marking for easy component and lead designation. DIP zone coordinates are marked along the board edges, while lead designations are etched into the power and ground buses.

8804 Plugbords are priced at $21.95 each in quantities of one to four; $19.76 each in quantities from five to 24; and $17.56 in quantities over 25.

Vector Electronic Company, 12460 Gladstone Ave., Sylmar CA 91342.

Wire-Wrapping Kit

Model WK-5B is a unique new wire-wrapping kit that contains a complete range of tools and parts for prototype and hobby applications, all conveniently packaged in a handy, durable plastic carrying case.

The kit includes Model BW-630 battery wire-wrapping tool complete with bit and sleeve; Model WSU-30, a remarkable new hand wire-wrapping/unwrapping/stripping tool; a universal PC board; an edge connector with wire-wrapping terminals; a set of PC card guides and brackets; a mini-shear with safety clip; industrial quality 14-, 16-, 24- and 40-pin DIP sockets; an assortment of wire-wrapping terminals; a DIP inserter; a DIP extractor and a unique three-color wire dispenser complete with 50 feet each of red, white and blue Kynar insulated, silver-plated solid AWG 30 copper wire.

Priced at $74.95, the WK-5B wire-wrapping kit is available from your local electronics distributor or directly from OK Machine and Tool Corporation, 3455 Conner Street, Bronx NY 10475.

Free Heath Brochure

Heath Company has recently issued a free computer facts brochure entitled "Why You Should Consider A 16-Bit Microcomputer." The 8-page brochure has information for those personal computerists who are undecided over the choice of an 8-bit or a 16-bit computer. The advantages of the 16-bit computer are discussed at length, as are the limitations of the 8-bit computer. Also covered are important topics such as computing power, software, service, support, reputation, quality and reliability.

Included also in the brochure is an introduction to the H11 Computer, Heath's 16-bit machine that utilizes the Digital Equipment Corporation LSI-11 CPU.

WK-5B kit.

The H11 is available both in kit form and as a completely wired and tested unit that is fully compatible with most DEC accessories and peripherals.

For a free copy of the new computer facts brochure, write Heath Company, Dept. 350-650, Benton Harbor MI 49022.

Foreign Language Programs

Musgrove Engineering announces the first of their Foreign Language Series educational programs, Foreign Language Vocabulary, a bidirectional program for the instruction, practice and testing of language-vocabulary skills. Languages offered include French, Spanish, Italian and German.

Program features include separate modes for vocabulary instruction, practice drills and testing, selectable by the user at any time during program operation. The user may also alternate language direction (English-to-French or French-to-English) to improve comprehension. The Educator option permits the creation of files for the storage of student identification, test responses and test scores for use in a classroom situation.

Foreign Language Vocabulary is written in BASIC for ease of adaptation to all microcomputer systems. Each volume includes an annotated program listing and program flowchart to assure ease of user loading and understanding. Single-statement lines are used to avoid confusion and to permit ease of user modification.

Each volume of Foreign Language Vocabulary is priced at $5, with the Educator option costing an additional $3. All four volumes are $17.50, $27.50 with the Educator option. Quantity discounts are available.

Musgrove Engineering, 9547 Kindletree Dr., Houston TX 77040.

Terrapin Turtle

Terrapin's computerized Turtle.

The Terrapin™ Turtle, a small electronic robot controllable by microprocessor, can "walk" (roll), touch (with its 3½-inch radius hemishperical dome) and draw (lowering its pen attachment) as programmed. It has lights for eyes and a speaker to emit sounds. The Turtle requires a parallel interface (one compatible with an S-100 bus is available as an accessory). Each Turtle comes with ten feet of cable and may be purchased either as a kit or fully assembled. Each kit comes with a tested, 20-page instruction manual.

The Turtle can be used to map rooms, solve mazes, teach simple geometry or programming concepts, as well as many other tasks. The Turtle is 5 inches high, crawls at 6 inches per second and is extremely versatile due to its touch sensors. Brochures are available. Kit $300; assembled $500; interface $40.

For further information contact: David McClees, Terrapin, Inc., 33 Edinborough St., Sixth floor, Boston MA 02111.

LETTERS

Thanks

When I was growing up I was taught to say "Thank you" whenever someone helped me in some way. This is a "Thank you" to you folks at *Kilobaud.*

For the past year, while I have been stationed with the Navy in the Philippines, I have been trying to learn about this new hobby of microcomputers. Your philosophy of keeping everything in plain, nontechnical English has been of great help, especially since I have no computer background and no one available locally to answer my questions.

I have read and reread every issue trying to learn as much as I could, and when that didn't work, I wrote letters to you at *Kilobaud* and to authors of several of the articles, making sure I enclosed a SASE each time. Each time I received a reply that was friendly, informative and encouraging. Judging by the responses I received, there is no danger of the people who have the experience already forgetting what it is like to be a beginner full of questions. To all those authors who answered my letters, I extend a warm "Thank you" also.

Some people deserve personal recognition. Assistant Editor Jeff DeTray took time out from his duties to write personal letters that cleared up some problems I was having at the time. And the l-o-n-g letter I received from author Ed Juge went a long way towards convincing me the TRS-80 is the one for me. Thanks, Ed.

The only complaint I have is not against the *Kilobaud* staff or the authors but against some of the advertisers. I sent five or six of the reader-service cards requesting information from about 25 companies. I received six replies. I also sent three personal letters to as many companies and received *no* replies. I did my best to keep track of which companies answered and which didn't, and plan to schedule any future buying plans accordingly.

I returned to the U.S. in July, to Kingsville TX, and can now pursue this exciting hobby at long last. And I hope I won't have to write so many letters.

So thanks again to everyone on the *Kilobaud* staff, and to Wayne Green for putting it all together so well.

Dan Lane
Kingsville TX

Cautionary Note on Batteries

I wish to comment on the article "Protect Your Memory" by Charles R. Carpenter in the March 1978 issue. I realize he was presenting only one possible solution to the "memory loss through power failure" problem, but the article does contain information that may be misleading and could possibly create a hazardous situation.

Mr. Carpenter states, "trickle charge the battery to keep it at maximum potential," but trickle charging storage batteries is recommended only for nickel-cadmium batteries. Both lead-acid and alkaline storage batteries require constant voltage sources for maintaining charge in standby service. Severe shortening of service life can result from the use of trickle chargers on both lead-acid and alkaline batteries.

Furthermore, if left on too high, lead-acid batteries present a potential hazard of trickle charge due to the generation of explosive gases and the possibility of corrosion or burns from escaping acid. Charging any storage battery and maintaining it in standby service requires that specific currents and/or voltages be applied, depending on the type and the capacity of the battery.

Perhaps you should warn your readers to not apply just any old voltage or any old current to the battery they may be using for memory save. Instead, they should follow the recommendations of the battery manufacturer regarding charging requirements for standby service. This is especially true if the battery is of the lead-acid (automotive) type. I know of instances when lead-acid

batteries have exploded while being charged, and in one case after the battery was off the charger for several hours. True, a spark or flame was present in each case to ignite the gases, but how many computerists smoke! I don't think that any computerist wants sulfuric acid splashed on himself, or on his computer, so a word of caution seems in order.

J. D. Martin
Pasco WA

Large Number of Comments on "Little Bits"

Your readers should be alerted to a serious flaw in the flowchart on page 69 of the July issue in the short piece by William Colsher titled "Develop Your Own Square Root Routine." The last step in the loop is given as:

```
LET R = 1/2(R/N+R)
```

It should read:

```
LET R = 1/2(N/R+R)
```

In addition to this error, which prevents the algorithm from giving the correct answer, values of N greater than 1,E16 will cause an overflow in many systems due to the test of R*R against N when R = N/2. There are further objections from the standpoint of efficiency and accuracy.

Curtis F. Gerald
San Luis Obispo CA

In addition to this letter from Curtis, we received five other letters, from Joe Ponder and R. L. Turner, Seattle WA; Allen Watson, Redwood City CA; Gordon Tillman, USS Tecumseh; and George Hill, Chevy Chase MD, all indicating they thought the algorighm was faulty.—Editors.

In reading "Little Bits" (*Kilobaud*, July 1978) I see a short article from Jack Starr regarding the need to improvise (?) when using SWTPC 8K BASIC. While the technique shown will work it should be noted that Example 1 will also work. These statements have the added virtue of closely resembling the form and function of the original statements. I regularly deal with 6 different implementations of BASIC on a regular basis and am not aware of any of those that cannot support the set of statements shown in the example.

```
102  IF Z<>13 THEN IF Z<32 THEN B1 = 1:GOTO 340
130  IF Z = 82 THEN IF Z = 1 THEN GOTO 180
```

Example 1.

The reason a problem (?) like this occurs at all is due to the ever increasing trend toward the implementation of interpreters that subvert widely used and understood BASIC syntax.

Joseph J. O'Loughlin III
Huron OH

Nearly the same response came from Ron Anderson, "an avid 6800 user" from Ann Arbor MI.—Editors.

Explanations, Please

It would seem to be time for an explanation of the aims, purposes and results of the *Kilobaud* Laboratory. This is prompted by Wayne's recent comments that "we've never managed to acquire a working 6800-based system" (Kilobaud Newsletter, 5/12/78. Similar remark in Publisher's Remarks, *Kilobaud* No. 18). The ads indicate that there are a number of companies manufacturing 6800-based systems, most of which have been around awhile. Our local computer club has had three different 6800 systems demonstrated, all of which worked. Two of our local stores sell 6800-based systems, and they report no great number of customer complaints; in fact, they say the biggest number of complaints has to do with getting delivery on accessories and expansion items, so the customers must like the 6800-based systems. So do us all a favor and tell us what your problems are.

I also refer to "Publisher's Remarks," No. 19: "A Need for Standards." It might possibly be advantageous to have a "standard bus," but is it really necessary? Most of the different bus designs available today seem to try to take maximum advantage of the processor signals available, *and the devices available to interface with it.* At least we are told that this is the reason Mits chose to use two unidirectional data buses even though the 8080 used a bidirectional data bus. Mention is often made of the "de-facto standard" nature of the Mits bus due to the number of boards available for it. This makes one wonder how many of these boards are available with supporting software that runs on a non-8080/Z-80 CPU, and how many could interface through a serial or parallel port rather than on the bus. A large number of the "S-100" bus boards are memory boards; is there really a problem getting memory boards for any bus structure?

How about a new column on "the ideal bus"; let your readers state their opinions on how it should be designed, and what features it should have. Since I am not a hardware type, my biggest question has to do with the technical aspects of on-board regulators vs large regulated power distribution on the bus.

R. S. Downs
Raleigh NC

As of this date we just haven't been able to arrive at an arrangement to obtain a 6800-based system. It hasn't been for lack of trying or interest, though. The laboratory itself is being used primarily to test programs for prerecorded cassettes . . . and eventually to teach uninitiated staffers how to program, and about buses, etc. Maybe you don't think a standard bus is necessary, R. S., but it sure would reduce the frustration level for many of us (see "Compatibility and the Altair Bus," Kilobaud *No. 19, page 100).—Editors.*

Due Credit

I noticed in the June 1978 issue ("ASCII to Baudot . . . ," p. 80) a circuit that looked familiar. Sure enough, after reading that it was first published in the March 1976 issue of *Radio-Electronics*, I realized that it was *mine!* I wonder why Maury Goldberg told me back in 1976 that he couldn't use my circuit. Could it have been to avoid the tiny royalty?

Had Mr. Lehman or Mr. Graham purchased the *original* kit from SWTP (at $24.50, not $35), they would have received a perfectly plated board, *all* parts and an offer from me to program the EPROMs as well as assist in any technical problems. I still provide this service, as anyone who provides a SASE can testify.

Incidentally, the circuit (as well as the Baudot to ASCII converter in the April 1976 issue of *Radio-Electronics*) has worked very well for the great number of those getting the original. The 7400 series ICs were chosen instead of UARTs, etc., to hold the cost of the kit below $25. I have seen more exotic converters and some with fewer ICs, but none below $24.50.

As for MMM, perhaps what they saved in royalties and technical help they lost in goodwill and sales. I guess all I can ask for now is due credit for the circuit.

Roger L. Smith
Mesa AZ

TRS-80: Query and Critique

I have recently become interested in microcomputer systems. I am basically a hobbyist and game player, but would also like a system to do home financial routines and other similar home functions.

I have considered several different systems starting from the 4K, Level I TRS-80, all the way to a built-up system costing upwards of $2000, and I have also written small programs for the Apple II.

I have settled on the 16K TRS-80; however, I do not wish to invest an additional $200 for the CRT. Is there any information available to aid me in the construction of an interface for the TRS-80 with an ordinary b & w television? Are there any serious disadvantages to doing this?

As I stated, I am mainly a game player. I am looking for a chess routine (simple 1, 2 or 3 moves ahead) to fit in the 16K memory of the TRS-80, written in BASIC.

Sanford A. Meyersfield, M.D.
Great Neck NY

It's rather surprising that no one has developed a TRS-80 interface to a regular television. (The Radio Shack has an opto-isolator built in so more than a simple TV modulator is required.) Maybe a reader can come up with something. Using T-BUG you could probably enter the 8080 version of Micro-Ware Ltd.'s Microchess (see Kilobaud *No. 20, p. 74 . . . if they provide source listings—we think they do), cost of which is $18. Micro-Ware, 27 Firstbrook Rd., Toronto, Ontario Canada M4E 2L2.—Editors.*

Wayne Green's remark on p. 7 of the June 1978 *Kilobaud*—"the recent blast in PCC at the TRS-80 is a mystery. . . . "—is itself a mystery. The TRS-80 has a great many shortcomings; some of the uglier ones are:

- The cassette recorder operates at a slower speed than most simi-

lar equipment. The recorder (to put it charitably) is awkward to use.

- The Level I BASIC is slow and primitive.
- The keyboard has no rollover.

The PCC article dealt at some length with these and other deficiencies. Neither Wayne nor Radio Shack (whose response to the criticism was printed in the same issue of PCC) addressed these substantive issues.

I think *Kilobaud* has an obligation to its readers to print the straight scoop about the drawbacks of equipment, especially that being offered to beginners. Reading Wayne's remarks in the June *Kilobaud*, one gets the impression that the TRS-80 and the PET are machines of equivalent quality and performance; this simply isn't so.

Dave Caulkins
Los Altos CA

Since I've had no problems using the TRS-80 recorder and letters from readers have been remarkably quiet about this, it is difficult to accept it as a notable problem. That Level II is a lot faster should still further complaints. I assume that just about everyone will be upgrading their systems to Level II.

Although Level I BASIC is slow, it is a darned good 4K BASIC, and it seems unfair to me to compare it with Level II or anything else other than Tiny BASICs, in which case it comes out well.

Rather than looking for things to criticize, I tend to look for the good aspects of equipment, making note of any serious shortcomings only. In the case of the TRS-80, there is no question that owners are wildly enthusiastic about it and tend to get upset over unfair criticisms.—Wayne

Another Dimension of "3-D Tic-Tac-Toe"

I really enjoyed the April article on 3-D Tic-Tac-Toe (page 66). However, when I programmed it on my machine, the computer took over eight minutes to make its first move. That was not acceptable, so I did some analysis of the algorithm and discovered a way to increase the speed of the computer moves.

The version of your article first calculates the value of each of the 76 possible winning combinations. It next proceeds to evaluate the value of a move in each empty box of the 64 possible boxes. The program does this for each box by searching the entire list of 76 winning combinations to see if that box is in each winning combination. If so, it adds the value of that combination to the sum for that box.

This procedure searches each of the four positions of the 76 winning combinations for each open box. This is an excessive amount (304 searches per box) considering that most boxes (48) are in only four winning combinations. The remaining 16 boxes (the eight outer corners and eight innermost boxes) are in seven.

My machine, Radio Shack TRS-80, Level I BASIC, took over eight minutes to calculate the first move under the author's design and under 75 seconds with my proposals.

The author's scheme of awarding points for various values of a possible winning combination (Table 1) is too conservative. Consider the following move:

Player	Computer
1,4	1,1
1,13	1,7
2,4	4,4
2,13	4,13
2,11	2,7
4,1	

The author's scheme will now move at 2,9, a value of 6. A move to 4,7 sets up two moves of 3 in a row 1,7 2,7 —, 4,7 and 4,4 4,7 —, 4,13. This move (4,7) has only a value of 4 under the author's weight scheme.

I propose the following modification to the points awarded for winning combinations (author's Table 1) (only changes are shown):

Value	Points Awarded
2	104
10	12

The rationale for this scheme is that we are searching for a move, such as 4,7 above, which places a third move for the player in two winning combinations where that player already has two moves. The result is the points for a box needed to ignore the tens digit if between 10 and 19 and the hundreds digit if between 100 and 199. This is summarized as:

if $10 \leq V \leq 19$ then $V = V - 10$
if $100 \leq V \leq 199$ then $V = V - 100$

Of course the computer must first block a player's chance to set up two combinations with three entries. That is the reason for the extreme weight of 100. With this proposed algorithm, box 2,9 has a value of 6 (106 reduced to 6), while box 4,7 has a value of 24.

The second modification consists of adding a small value (I chose .1) to all boxes of a winning combination with no moves in it. The purpose of this is to aid tie breaking. Given equal blocking and scoring power, it is better to move to the box that contributes to the most unused combinations. The following game shows this:

Player	Computer
1,1	1,2*
1,6	1,11
2,6	3,11
2,11	3,6
3,5	3,9
1,5	1,9
2,5	4,5
3,7	4,8
2,7	2,8
1,7	4,7
1,8: wins	

The computer's first move (1,2) is a strategical error. All moves that block the seven winning combinations involved with 1,1 produce the same value. The computer assigns the value of 1 to box 1,2 first and discards ties after that. Consequently, a move is selected that blocks one of the player's chances of winning, but only provides 3 chances for the computer to win. This is not as good as 1,4; 1,13; 1,16; 4,1; 4,4; 4,13; 4,16; 2,6 or 3,11, which also block the player's move but open six winning chances for the computer. With this change the computer moves to 2,6.

In summary, I enjoyed the author's article, but think that the above changes will run faster and make the computer do even better, as if it needs any help.

Robert E. Heath, II
Woodbridge VA

Congratulations

There seemed to be a number of requests for reader feedback in the June 1978 issue, and I thought I would reply.

The "Tour of the Faire (Part 1)," with its many pictures of the people behind the products, was well worth the space devoted to it. Since much of microcomputer purchasing is still a mail-order business, this type of article provides a vital psychological link to the people involved in the many companies advertising in *Kilobaud*.

Speaking of advertising, I don't think I am alone in that the first thing I read when I get my copy of *Kilobaud* is the advertising. I don't find the ads an intrusion but a service. I am educating myself with both the articles and the advertising, and when I purchase microcomputer supplies they are chosen from among the regular *Kilobaud* advertisers.

Congratulations for publishing the consumer report on Mini Micro Mart as well as the information on Norman Henry Hunt. Both articles indicate a concern for your readers that goes beyond customary publishing practice. I tried to return the favor in a small way by mentioning *Kilobaud* in a recent order to one of your advertisers.

Tom Vollmer
Honolulu HI

Unregal Royalties?

I would like to make a few remarks about the Publisher's Remarks (June 1978). Although there are several software distributors, it would be good for *Kilobaud* to enter the field. However, the example given shows the ridiculousness of giving the programmer a flat 20 percent royalty, since although he would get $859,680, *Kilobaud* would get $3,438,720. I would not want to buy a program written by anybody stupid enough to fall for that. Twenty percent is a reasonable royalty for inexpensive programs where there is not much profit for the distributor, but it is unreasonable for more expensive packages.

It was surprising to learn of the large number of systems that *Kilobaud* has. Why haven't you given us some articles comparing them? Many writers seem to have had no experience with systems other than their own and so it is hard to tell if their expectations have anything to do with reality. You seem to be very well equipped to write some excellent product reviews.

I have also gotten tired of hearing Wayne Green bad-mouth *Byte*. I know that he feels that he has had a raw deal, but why take it out on us? Like most of the other readers of *Kilobaud* I prefer it to *Byte* so he is talking to the wrong audience.

Please don't think that I have anything against Wayne Green. His remarks are always entertaining and sometimes very enlightening. I would just like to ask him to ease off on his comments about the competition.

Ralph Johnson
Galesburg IL

In the example, Kilobaud *would gross $3,438,720, but then there are: start-up costs (which on a relatively large scale are considerable), promotion, packaging, labor (it takes quite a few people quite a bit of time to get the cassettes into final form) . . . and,*

finally, taxes to consider. Furthermore, how many programwriters have access to the equipment and personnel to produce, promote and market their own software (on a relatively large scale)? We have a number of systems, and comparison articles is a good idea. Everyone here is very busy (although staffer Steve Lionel did find time to write a fine review of the Heath H11, p. 52)—maybe if someone could stop by . . . —Editors.

Evolutions

I was delighted to see my article "Memory Troubleshooting Techniques" (*Kilobaud*, October 1977) followed up by Rod Hallen's excellent article in the July 1978 issue of *Kilobaud* (page 70). Mr. Hallen has presented a relatively painless implementation of my flowchart. In that article he mentioned that he artificially introduced some shorts and opens in order to simulate faults. Some readers may think that such faults may rarely occur. It has been my experience that such faults are fairly common with homemade printed circuit boards. While most people buy their processor and memory boards, there are still a lot of homemade PC boards out there used with computers in one way or another.

In my original article I made a few comments about these PC board problems and as a result I have had numerous inquiries about how to make defect-free prototype boards. In fact, just that brief mention of PC boards in general brought responses that indicated that people fairly knowledgeable in electronics were in desperate need for someone to make boards for them. To make a long story short, that article led me in a totally unexpected direction: the prototype printed circuit business. As expected, most customers are computer enthusiasts.

Thank you *Kilobaud*.

Charles E. Cook
Refugio TX

Tom's PROM

I have had one inquiry about the PROM Programmer by Tom Hayek (No. 19, page 94) concerning programming other PROMs. A cursory look indicates that the circuit and board will work for a number of other PROMs.

I have listed them in a table to show what changes have to be made. The Harris PROM requires a high on pins 20 and 21, while the 82S series requires a low; this is all programmable. The intial bit pattern (INBP) is low.

It will even work with an 82S 190 and 82S 191 with pin 21 brought out for addressing (2048 x 8).

Perhaps there might be some people who can use this information.

Ozzie Stafford
Greensboro NC

PROM	Pin 20,21 $\overline{CE}$	Collec Resistors	Int Bit Pattern	Mfg.	Comment
82S140/141	0,0	10K	0	Signetics	Program units in
82S180/181	0,0	10K	0		256 byte jumps
82S2708	0,0	10K	0		
82S190/191	0,X	10K	0		Address pin 21
7641-5	1	560Ω	1	Harris	
7681-5	1	560Ω	1		
				TI	N.A.

A "Best Buy"

I recently purchased the "Black Box" printer advertised by Expandor Incorporated, 400 Sainte Claire Plaza, Upper St. Claire PA 15146. The 80-column impact printer is available with a parallel interface and controller at $396. It mates perfectly with the parallel I/O card in the Heath H8 computer. Ten minutes to wire a 30-pin plug is all it takes to be up and running.

Delivery time from Expandor was approximately four months. However, at $396 the wait was worth it. This must be rated a "best buy."

Laird D. Schearer
Boulder CO

KB CLUB CALENDAR

Steve Fuller

Arlington MA

A free newsletter is available to TRS-80 users from the TRS-80 Club, 96 Dothan St., Arlington MA 02174. A software library is also available for reproduction cost.

Belvidere IL

Here's an update from the Blackhawk Bit Burners Computer Club: According to club president Frank Dougherty, the Bit Burners now have 40 active members from the Rockford area.

If you'd like more information, write to the club at 325 Beacon Drive, Belvidere IL, or call (815) 544-5206 evenings.

Lynn Haven FL

Don Palmer of the Panama City Computer Society would like some assistance in organizing his new club. He asks if there is a standard charter that would be compatible with some existing clubs, and is also interested in any national affiliation to which his club may apply. If you can help, or would like to join the club, write to Don at 815 Ky Ave., Lynn Haven FL 32444.

Philipsburg PA

PROGRAM, a club for PET owners, has been formed to address the need for programs and consumer/hobbyist education. For an annual fee of $27 members receive a monthly cassette containing PET-related articles, applications, hardware sources, reviews of new peripherals and programming hints. Also included are several games, and a variety of household and business programs.

Write to PROGRAM, PO Box 461, Philipsburg PA 16866.

Sacramento CA

TRS-80 owners in the Sacramento area are invited to call Sal Alestra at (916) 927-0237 for information concerning a new users group.

Chicago IL

William Colsher sends word of a new Digital Group organization in Chicago.

"The Digital Group Group of Chicago was formed to provide a forum for the exchange of ideas, software, fixes, etc., by owners of Digital Group computer systems.

"We meet on the last Tuesday of each month in the meeting room of Consumer Systems, 2107 Swift Road, Oak Brook IL. Meetings are held at 7:30 PM.

"Annual membership dues of $5 are used to help defray the costs of producing the club newsletter and monthly meeting notifications. The newsletter is currently running about 4 to 6 pages and contains news of club activities, Digital Group compatible hardware and software, as well as articles and reviews by club members."

For more information write The Digital Group Group of Chicago, c/o William L. Colsher, 4328 Nutmeg Lane, Apt. 111, Lisle IL 60532, or call club president Rich Kurtz at 852-5772.

Rome NY

Rome Area Computer Enthusiasts (RACE) was recently formed here for computerists in the Rome-Utica area of central New York.

The club meets on the second Tuesday of each month at Patty's Stagecoach Inn at 7:30 PM. Special interest groups have been formed for beginners and for owners of 6800 and 8080/Z-80 microprocessors. A newsletter, *Micros Along the Mohawk*, is a regular publication of the club. Interested hobbyists can write to Mike Troutman, RD 1, W. Carter Road, Rome NY 13440, or call him at (315) 336-0986.

This column is available for you to report on your club's activities such as regular meeting schedules, special events or programs, swap meets or any endeavor that will be of interest to your fellow hobbyists. If your announcement contains timely information, send it at least two months prior to the date(s) mentioned in the announcement.

Kilobaud Club Calendar
c/o Steve Fuller
334 Sterling St. Unit A-3
West Boylston MA 01583

BOOKS

(from page 8)

program which will do the same job that would otherwise require all those integrated circuits. The program is not a particularly efficient one, however, and we learn how to clean it up in the next section, reducing it to 105 lines. It should be noted that the line count of 105 includes a considerable number of explanatory comments. If the comments were deleted, the program would be quite a bit shorter, but nearly impossible for anyone unfamiliar with it to decipher at a later time. A third chapter is devoted to streamlining the program even further.

The book concludes with an examination of the Z-80 instruction set and a look at some commonly used subroutines.

While you shouldn't throw away your soldering iron or wire wrap tool just yet, the techniques explained in this book could be used to perform some jobs now done with hardware. If your bag is "doing it with software," you ought to take a look at *Z-80 Programming for Logic Design.*

Jeff DeTray
***Kilobaud* Staff**

***Microprocessor Interfacing Techniques*, Second Edition**
Austin Lesea, Rodnay Zaks
Sybex, Inc.
Berkeley CA, 1978

Microprocessor Interfacing Techniques provides a highly readable compilation of basic interfacing techniques and standards that must be understood before any wires are connected to a computer. The book provides both general and specific information for interfacing a micro to anything from a keyboard to a floppy disk. Also covered are parallel standards such as the IEEE 488 bus and the S-100 bus along with some clever insights on why standards are standards. Serial standards such as the RS-232 C and the various synchronous data link controller (SDLC) schemes are introduced.

Before designing any interface, the designer should at least know of the existence of these standards and realize that it isn't necessary to start from scratch when designing an interface. The major advantage to be gained by understanding the interface standards is that the final system may be signal and plug compatible with other systems.

Another area given considerable coverage is that of analog interface. Almost all naturally occurring signals are analog in nature, consequently any control or monitoring functions to be performed by a microprocessor will require some amount of analog interface. The authors of this book have provided all the background information needed to design a data acquisition system such as sampling, multiplexing, analog to digital conversion and digital to analog conversion. As microprocessors are increasingly turned to real-world problems, analog interfacing techniques become more important.

Microprocessor Interfacing Techniques contains a considerable amount of useful information in a form that is easy to read and understand. This book should be read by anyone who is in the preliminary stages of interface design and does not have a lot of preconceived ideas about what the interface will be like. Although interface standards can sometimes be unwieldy, there are a lot of good reasons for using standard interfacing techniques.

David G. Herold
Athens OH

PUBLISHER'S REMARKS

(from page 7)

Home Work, Too

My plan for producing software is to first have an editor check each submitted program for completeness, good documentation, non-conflict with other programs in production or published, sale possibilities, etc. If the program passes this preliminary hurdle we will have a person who makes up ten copies of the program and the documentation and sends these out to ten associate editors for thorough review. This job will mean the editor will have to have an appropriate system at home and the background to do a good evaluation of the program. We'll expect a complete critique of the program plus suggestions for changes, improvements, variables that users might want to change, suggestions for customization of the program, etc. This work will be paid on a per-hour basis . . . starting at $3 per hour. This is low for in-house work, but for work done at home at the convenience of the editor—and for work that is hardly work—I suspect we'll have a good choice of associate editors. Is this something you could do well and would be fun for play-work in your spare time? This is a chance to get in on the ground floor of an exciting new industry.

If you'd like to get a tryout as an associate editor of our software, drop a line to Wayne Green, *Kilobaud*, Peterborough NH 03458, and give me your background, hardware, etc. We'll be needing editors with TRS-80 4K, 8K and 8K Level 2 systems . . . plus PET owners . . . and most of the other type of systems. Considering the number of TRS-80 systems extant, we'll be needing TRS-80 editors in goodly numbers.

Extra Carrot

Yes, $3 per hour is a pittance for top-notch programmers. They should get ten times that, no doubt. In order to equalize this pay a bit and make the project more attractive to programmers . . . and to encourage first-rate work . . . we will be setting aside an extra 2 percent royalty on sales and splitting this with the associate editors who contribute the most toward the success of a program.

This means that an editor who might be splitting this 2 percent with one other editor would do quite well on a sale of 50,000 of the program. At a retail price of $7.95 for the program and a wholesale gross of $4.293, this would bring each of the two editors $2146.50 in royalties. You won't buy too big a yacht with that, but you can buy one hell of a rowboat . . . maybe even an outboard.

I am hoping that high pay for programs and for editing will result in a large quantity of first-rate programs . . . and that this will result in the spread of microcomputing into every corner of the world.

What type of programs? Just about anything . . . games are always popular, of course, but as the well-known games are all published there will be a need for the development of new games, and we may find them selling even better for computer use than do the board games. There is a need for hundreds of business programs . . . home programs . . . educational programs . . . medical . . . the list is really endless.

Submitting Programs

As the number of programs submitted grows, so will the requirements for perfection and detail. If you have a program that you think might be worth publication on cassette and that you think might sell in quantities of 50,000 or more over the next year or so, you really should get it into as good shape as you can and submit it to us for possible publication.

The royalties are substantial and should keep you going for several weeks, at least. The royalty is 20 percent of the wholesale

(continued on page 118)

Reader Responsibility

One of your responsibilities, as a reader of *Kilobaud*, is to aid and abet the increasing of circulation and advertising, both of which will bring you the same benefit: a larger and even better magazine. You can help by encouraging your friends to subscribe to *Kilobaud*. Remember that subscriptions are guaranteed—money back if not delighted, so no one can lose. You can also help by tearing out one of the cards just inside the back cover and circling the replies you'd like to see: catalogs, spec sheets, etc. Advertisers put a lot of trust in these reader requests for information. To make it even more worth your while to send in the card, a drawing will be held each month and the winner will get a lifetime subscription to *Kilobaud*!

George Tucker of Janesville WI is the most recent winner of a lifetime subscription to *Kilobaud*.

(Con)text Editor

Here's something different—a text editor in BASIC. It's a full-fledged editor; you'll like it.

R. M. Law
D. C. Mitchell
5521 Woodward
Downers Grove IL 60515

Almost everyone uses some kind of editing to create and debug new software. Whether the task consists of merely deleting and inserting lines in BASIC, or whether it encompasses writing letters and forms for business, a good context editor is handy.

This article presents a context editor which incorporates many of the features found on large computer systems. Furthermore, it is written in BASIC rather than assembly language, which makes it machine independent and easier to understand and modify. The drawback of writing an editor in a high-level language, of course, is its slow speed. Thus, our main program-design criterion was to reduce as much processing time as possible. The result represents a useful trade-off among editing features, speed and memory requirements.

In our home, the editor receives constant use for record keeping, letter writing, recipe collecting, record and tape cataloging, name and address file updating and retrieving, daily calendar and message reminding and countless other applications where information is stored in text form. The editor was also used to prepare this article for publication. We hope you will find it a useful addition to your program library.

BOTTOM	Move the current-line pointer to the last line of the data set.
CHANGE	Change text on a line or a range of lines.
DELETE	Delete one or more lines from the data set.
EXCERPT	Duplicate a specified number of lines at somewhere else in the data set.
FIND	Find the first occurrence of specified text.
INSERT	Insert new lines into the data set.
KILL	Delete the entire data set.
LOAD	Get a data set from tape.
MOVE	Move a specified number of lines to somewhere else in the data set.
NEXT	Move the current-line pointer forward or backward a specified number of lines.
OVERLAY	Delete one or more lines and insert new lines in their place.
PRINT	Print a line or a range of lines with their corresponding line numbers.
QUIT	Exit the context editor.
RENUMBER	Reorganize the data set sequentially by integer line numbers.
SAVE	Put the current editor data set on tape.
TOP	Move the current-line pointer to the first line of the data set.
WRITE	Print the entire data set in its final form without line numbers.
XRAY	Used for debugging.

Fig. 1. Editor commands.

The Editor: Makeup and Use

The entire editor occupies 16K of memory including Mits 8K BASIC Version 4.0. It is designed to hold one entire page of text at a time—a page is defined as 60 lines of 72 characters (this can be easily adjusted to other values). The editor can load, edit and store these pages. Besides text, each line is paired with a line number that allows the user to reference desired lines. Fig. 1 shows the complete command repertoire of the editor.

A context editor differs from a text editor in that it allows the user to change the contents of a line, or a range of lines, by typing only the text to be changed and the replacement text. It can also locate text occurrences anywhere in the data set without having to be told the line where the text resides.

The editor commands themselves offer a great deal of freedom in how the user enters them, and what options are present. All commands can be abbreviated, and many parameters are optional. For example, to delete lines 3 through 5 in the data set, the user could enter any of the following:

```
DELETE 3-5
D (3,5)
D3/5
DEL 03 05
DELETE 3.0:5.0
```

The context editor will try to make sense out of many individual styles of users. The general rules of syntax for each command are in the Editor Command Guide. In addition to these commands, there are several main concepts with which to become familiar.

Line Numbers. If you have not used an editor with line numbers, you will appreciate the time-savings gained in locating or referencing items in the data set. In addition, line numbers provide for fewer errors in deleting and moving, since you always know exactly where you are steering the commands.

The context editor permits fractional line numbers. Thus, the user can specify an increment that is to be added to the

line number of the previous line to produce the current line number. This increment is initially set to 1, but can be changed in the INSERT, MOVE, OVERLAY and EXCERPT commands to be anywhere from .0001 to 10,000.

Before inserting a new line, the editor verifies that the new line number is not greater or equal to the line number of the following line. If it is greater or equal, an error message will occur, indicating that you are trying to insert or move on top of existing lines.

The context editor also freely permits the use of nonexistent lines numbers—a feature that few others offer. For example, if you enter DELETE 4-6, it is not necessary that lines 4 and 6 actually exist. The editor will lock onto any lines that do exist in that range and delete them.

Similarly, you can INSERT 3 even though line 3 doesn't physically exist, and the editor will begin numbering your new lines as if the line were there. In fact, the only time the editor won't let you operate on phantom lines is if the entire range you specify is nonexistent (at which time the operation is meaningless).

Although perhaps not obvious, this feature can save you a lot of work in certain types of situations, especially when you want to renumber or resequence only a portion of the data set.

Line numbers are limited to the floating-point precision offered by your BASIC interpreter; for Mits BASIC Version 4.0, this is six digits. Thus, line numbers such as 10000.1 and .987654 are legal, but numbers such as 10000.12 and 1.987654 will be truncated by BASIC when you attempt to use them.

Current Line. The context editor maintains a current-line concept for convenience. The current line is usually always the last line processed by the previous command. If you don't specify the line numbers in a command, the editor will assume you want to process the current line.

For example, if you type only DELETE, the editor will delete

```
CONTEXT EDITOR                               VERSION 5.0


?INSERT                              User is creating a new data set.
>?THIS IS LINE A
>?THIS IS LINE B
>?THIS IS LINE C
>?THIS IS LINE D
>?                                   User defaults to leave insert mode.
4    THIS IS LINE D                  System prints current line.
?PRINT 1-4                           User wants to print lines 1-4.
1    THIS IS LINE A
2    THIS IS LINE B
3    THIS IS LINE C
4    THIS IS LINE D
?INSERT 2 BY .1                      User wants to insert between lines 2 & 3.
>?THIS IS LINE E
>?THIS IS LINE F
>?                                   User defaults to leave insert mode.
2.2 THIS IS LINE F                   System prints current line.
?PRINT ALL                           User wants to see entire data set.
1    THIS IS LINE A
2    THIS IS LINE B
2.1  THIS IS LINE E
2.2  THIS IS LINE F
3    THIS IS LINE C
4    THIS IS LINE D
?DELETE                              Delete the current line.
3    THIS IS LINE C                  System backs up to previous line
                                     and makes it current.
?NEXT -1                             User backs up one line.
2.2  THIS IS LINE F
?CHANGE/LINE F/LINE H/               User wants to change text.
2.2  THIS IS LINE H
?TOP                                 User wants top of data set.
1    THIS IS LINE A                  System prints current line.
?FIND/LINE E/                        User wants to find text.
2.1  THIS IS LINE E                  System prints current line.
?C/LINE E/LINE G/                    User wants to change text.
2.1  THIS IS LINE G                  System prints current line.
?P ALL                               User wants to see entire data set.
1    THIS IS LINE A
2    THIS IS LINE B
2.1  THIS IS LINE G
2.2  THIS IS LINE H
3    THIS IS LINE C
?EXCERPT 1 2 3 BY .01                User wants to duplicate lines 1&2 after 3.
3.02 THIS IS LINE B                  System prints current line.
?NEXT -1                             User backs up one line.
3.01 THIS IS LINE A                  System prints current line.
?N -1                                User backs up one line.
3    THIS IS LINE C                  System prints current line.
?P ALL                               User wants to see entire data set.
1    THIS IS LINE A
2    THIS IS LINE B
2.1  THIS IS LINE G
2.2  THIS IS LINE H
3    THIS IS LINE C
3.01 THIS IS LINE A
3.02 THIS IS LINE B
?OVERLAY 2                           User wants to replace line 2.
>?THIS IS LINE X
>?THIS IS LINE Y
>?THIS IS LINE Z
>?                                   User defaults to leave insert mode.
1.03 THIS IS LINE Z                  System prints current line.
?P ALL                               User wants to see entire data set.
1    THIS IS LINE A
1.01 THIS IS LINE X
1.02 THIS IS LINE Y
1.03 THIS IS LINE Z
2.1  THIS IS LINE G
2.2  THIS IS LINE H
3    THIS IS LINE C
3.01 THIS IS LINE A
3.02 THIS IS LINE B
?RENUMBER                            User wants to renumber data set.
1    THIS IS LINE A                  System prints current line.
?P ALL                               User wants to see entire data set.
1    THIS IS LINE A
2    THIS IS LINE X
3    THIS IS LINE Y
4    THIS IS LINE Z
5    THIS IS LINE G
6    THIS IS LINE H
7    THIS IS LINE C
8    THIS IS LINE A
9    THIS IS LINE B
?KILL                                User wants to delete entire data set.
DO YOU REALLY WANT TO KILL THE DATA SET (Y/N)? Y
?P ALL                               User wants to see entire data set.
THE DATA SET IS EMPTY
?QUIT                                User wants to leave editor.
NORMAL END
```

Fig. 2. Sample run.

Editor Command Guide

BOTTOM

Locate and print the last line of the data set. The bottom line will become the current line. BOTTOM followed by INSERT will insert new lines at the end of the data set.

CHANGE <A< B>>*TEXT1*TEXT2*< PRALL>
< ALL>

Change the first occurrence of TEXT1 to TEXT2. If line numbers A and B are not specified then change TEXT1 on the current line. If only line number A is specified then only line A will be changed. If line numbers A and B are specified then the first occurrence of TEXT1 on each line in the range A to B will be changed. If optional parameter ALL is specified, all occurrences of TEXT1 on the current line or range of lines will be changed. Only the new current line will be printed. Optional parameter PRALL is the same as parameter ALL except all occurrences of TEXT1 will be printed. TEXT1 and TEXT2 must be surrounded by delimiter symbols (shown above as *) which can be any nonalphanumeric character, except, of course, those appearing in TEXT1 or TEXT2.

DELETE <A< B>>

Delete one or more lines from a data set. If line numbers A and B are not specified, the current line will be deleted. If only line number A is specified, only line A will be deleted. If line numbers A and B are specified, the range of lines A through B will be deleted; the current-line pointer will move to the line above the deleted range. If the first line of the data set is deleted, the current-line pointer will move beyond the deleted range.

EXCERPT <A< B>> C< BY N>

Duplicate line A or lines A through B to after line C. The original line A or range of lines A through B remain unchanged. If line numbers A and B are not specified, then the current line will be duplicated after line C. The increment amount for line numbers inserted after C can be changed with the BY N option.

FIND <A< B>>*TEXT1*< PRALL>

Find the first occurrence of TEXT1. If line numbers A and B are not specified, then FIND will begin with the current line and search until it locates TEXT1 or until the bottom of the data set is reached. If only line number A is specified, only line A will be searched. If line numbers A and B are specified then the range of lines A through B will be searched for the first occurrence of TEXT1. FIND will stop and print at each occurrence of TEXT1. To find the next occurrence, merely hit RETURN. The editor will continue searching until it finds the next occurrence. You can repeat this as many times as you wish. If optional parameter PRALL is specified, all occurrences of TEXT1 on the current line or range of lines will be found and printed. TEXT1 must be surrounded by delimiter symbols (shown above as *) which can be any nonalphanumeric character, except, of course, those appearing in TEXT1.

INSERT <A>< BY N>

Insert after line A, or after the current line if A is not specified. If A is 0 then insert above the first line of the data set. The editor keeps prompting for new lines of text. To leave the insert mode, enter a null line. The increment value for insertion can be set with the BY N option. The current line becomes the last line inserted.

KILL

Destroy the existing data set and restart the editor. This is useful after you have saved your data set on tape and wish to begin creating a new one.

LOAD

Load a data set from tape. LOAD does an automatic KILL. To begin loading, advance tape to where the page was saved and type LOAD.

MOVE <A< B>> C< BY N>

Move line A or lines A through B to after line C. The original line A or range of lines A through B is deleted. If line numbers A and B are not specified, then the current line will be moved to after C. The increment amount for line numbers inserted after C can be changed with the BY N option.

NEXT < N>
<+N>
<−N>

Move the current-line pointer + forward or − backward N number of lines. If N is not specified, the default is N = 1. The current-line pointer will be moved to the line following the current line. If +N exceeds the bottom of the data set the current-line pointer will be set to the last line of the data set. If −N exceeds the top of the data set, the current-line pointer will be set to the first line of the data set. In each case the new current line will be printed

OVERLAY <A< B>>< BY N>

Delete one or more lines and enter insert mode. If line numbers A and B are not specified, the current line will be deleted and insert mode entered. If only line number A is specified, only line A is deleted and insert mode is entered. If line numbers A and B are specified, the range of lines A through B is deleted and the insert mode is entered. Insert mode is terminated by a null line. The last line inserted becomes the current line. The increment value for insertion can be set with the BY N option.

---

the current line. Similarly, INSERT, without any line numbers specified, will insert immediately after the current line. This feature allows you to enter shorthand commands for frequently used operations.

If you are ever unsure of the current line, just type PRINT. However, the editor will ensure that the current line is always the previous line printed, and will generate automatic print calls wherever it is not obvious what line is current.

Delimiters. Delimiters are nonalphanumeric characters used to surround text in CHANGE and FIND commands. The user must choose characters that are not present in the text itself.

For example, if the user had the line THIS IS A SAMPL LINE, he could correct it to THIS IS A SAMPLE LINE by the command:

CHANGE/PL LINE/PLE LINE/
(SLASHES AS DELIMITERS)

but not by:

CHANGE PL LINE PLE LINE
(BLANKS AS DELIMITERS)

since blanks are part of the text. However, he could perform the same correction by:

CHANGE PL PLE
(BLANKS AS DELIMITERS)

since the blanks are not part of the text in this example.

Fig. 2 illustrates these principles through a sample run.

At the heart of the data set is a group of four matrices shown in Fig. 3. For every line I of text, there is an associated text entry L$(I), a line number L(I), a forward link to the next successive line number F(I) and a backward pointer to the previous line number B(I). Associated with these four matrices are two unique pointers: USED and UNUSED.

The used pointer points to the first (lowest) line number in the data set; the unused pointer points to the first vacant slot in the matrix, or the next place to put a newly inserted line. Thus, if USED = 0, the data set is empty; if UNUSED = 0, the data set is full. A zero entry in the F or B matrix indicates the end of a forward or backward chain, respectively.

Fig. 3 shows how the matrices are initially set up for a new

PRINT <A< B>>< ALL>

Print a set of lines, along with their line numbers. If line numbers A and B are not specified, then print the current line. If only line number A is specified, then line A will be printed. If line numbers A and B are specified, the range of lines A through B will be printed. If optional parameter ALL is specified, the entire data set is printed. The last line printed becomes the current line.

QUIT

Leave the context editor and return to basic command mode.

RENUMBER

Reorganize the data set sequentially by integer line numbers.

SAVE

Save the data set on tape. Start tape before you hit RETURN.

TOP

Locate and print the first line of the data set. The top line will become the current line. TOP followed by INSERT will insert new lines following the first line of the data set. INSERT 0 will insert new lines above the top of the data set.

WRITE

Same as PRINT except the line numbers are not printed. The WRITE command prints the entire data set, and is generally used to output the final form of a letter, document, etc.

XRAY

Console tool used to display all internal links and pointers associated with the data set.

```
UNUSED:   1            INITIAL DATA SET
USED:     0
```

I()	F()	B()	L()
1	2	0	0
2	3	1	0
3	4	2	0
4	5	3	0
5	6	4	0
6	7	5	0
7	8	6	0
8	9	7	0
9	10	8	0
10	0	9	0

Fig. 3. Four matrices.

data set; note that there are no entries on the used chain, and the unused chain fills the entire data set. Only ten entries are shown in the figure. In the actual code, there are 60 entries, one for each line of a standard Teletype page.

INSERT. Fig. 4 illustrates the algorithm employed for inserting a new record into the data set. First, a check is made to see if there is any free space in the editor buffer. If there are no vacant lines, then UNUSED = 0, and an error message that the editor buffer is full is forwarded to the user. If space is available, a new line is reserved for the new record by seizing the first free record on the unused chain [C = UNUSED] and by decreasing the size of the unused chain by one line [UNUSED = F(C)].

Now that the physical space exists for the new record, we must locate where the new line should be linked in order of its line number. That is, we must eventually arrange the used chain so that it is numerically ascending by line number.

Of course, one way to do this is to apply a sort routine to the entire data set to sort it lowest to highest line number. However, even with the very best sort routine, it is very slow compared to leaving the lines randomly placed but linking pointers to sequentially link the data

Program listing.

```
1 CLEAR3750:I=0:J=0:K=0:P=0:C=0:B=0:F=0:R=0:U=0:UN=1:S=0:E=0:M=0:L=0:N=0
2 Q=60:DIMF(Q),B(Q),L(Q),L$(Q)
3 B$=" ":A$="A":Z$="Z":X$="0":Y$="9":D$=".":C$="DIPCFONTBRMEWLSKOX"
5 PRINTCHR$(26)"CONTEXT EDITOR"TAB(50)"VERSION 5.0":PRINT:PRINT:PRINT
6 POKE2507,34:POKE2509,34
10 FORI=1TOQ:F(I)=I+1:B(I)=I-1:L(I)=0:L$(I)="":NEXT:F(Q)=0
11 U=0:UN=1:M=0:BY=1
25 L=L(M):P=1:I$=B$:INPUTI$:GOSUB1000:IFR<0THEN40
26 T$=LEFT$(T$,1):FORO=1TO16:IFMID$(C$,O,1)<>T$THENNEXT:GOTO200
30 ONOGOTO50,100,150,200,250,300,350,400,450,500,550,600,650,700,750,800
31 ONO-16GOTO850,900
40 IFO=5THENB=0:GOTO235
41 GOTO2000
50 S=L:E=L:GOSUB1000:IFR=2THENGOSUB1000:IFR>0THEN2001
51 IFR=0THENS=VAL(T$):E=S:GOSUB1000:IFR=2THENGOSUB1000
52 IFR=0THENE=VAL(T$):GOSUB1000:IFE<STHEN2004
53 IFT$="BY"THENGOSUB1000:BY=VAL(T$):GOTO55
54 IFR=0ORP=1THEN2001
55 IFBY<1E-4ORBY>1E4THEN2004
60 B=0:F=0:C=U:IFC=0THEN2006
61 IFC>0ANDL(C)<STHENC=F(C):GOTO61
62 IFC>0ANDL(C)<=ETHENB=C
63 IFC>0ANDL(C)<=ETHENF=C:L(C)=0:C=F(C):GOTO63
70 IFB=0ANDF=0THEN2002
71 M=B(B):IFU=BTHENU=F(F):B(U)=0:GOTO73
72 F(B(B))=F(F):B(F(F))=B(B)
73 F(F)=UN:B(UN)=F:UN=B:M=B(B)
90 IFM=0THENM=U
91 IFO=1THEN2100
92 I=S-BY:GOTO110
100 I=L:GOSUB1000:IFR>1THEN2001
101 IFR=0THENI=VAL(T$):GOSUB1000
102 IFT$="BY"THENGOSUB1000:BY=VAL(T$):GOTO104
103 IFR>=0THEN2001
104 IFBY<1E-4ORBY>1E4THEN2004
110 B=0:F=U:J=I+BY
111 IFF>0ANDL(F)<JTHENB=F:F=F(F):GOTO111
120 I=VAL(STR$(I+BY)):C=UN:IFO<>12THEN124
121 IFR=0ORL(R)>ETHEN2100
122 I$=L$(R):R=F(R):GOTO125
124 I$="":INPUT">";I$:IFI$=""THEN2100
125 IFUN=0THEN2003
126 UN=F(C):IFF>0ANDI>=L(F)THEN2005
127 IFB=0THENU=C:GOTO129
128 F(B)=C:B(C)=B
129 F(C)=F:B(F)=C:B(UN)=0:L(C)=I:L$(C)=I$:M=C
130 B=C:GOTO120
150 S=L:E=L
151 IFU=0THEN2006
152 GOSUB1000:IFT$="ALL"THENS=0:E=1E5:GOTO160
153 IFR>0THEN2001
154 IFR=0THENS=VAL(T$):E=S:GOSUB1000:IFR=2THENGOSUB1000:IFR>0THEN2001
155 IFR=0THENE=VAL(T$):IFE<STHEN2004
160 C=U:J=0
161 IFC>0ANDL(C)<STHENC=F(C):GOTO161
170 IFC=0ORL(C)>ETHEN180
172 J=1:PRINTL(C)TAB(10)L$(C):M=C:C=F(C):GOTO170
180 IFJ=0THEN2002
182 GOTO25
200 F=0:C=0:B=0:J=LEN(I$):R=J:I=0:IFO=5THENB=J
201 T$=MID$(I$,J,1):IFT$<A$ORT$>Z$THEN207
```

```
202 T$=MID$(I$,J,1):IFT$>=A$ANDT$<=Z$THENJ=J-1:IFJ>0THEN202
203 IFJ=0THEN2001
204 T$=RIGHT$(I$,R-J):IFT$="PRALL"THENI=2:GOTO207
205 I=1:IFT$<>"ALL"THEN2001
207 T$=MID$(I$,J,1):IFT$>=X$ANDT$<=Y$THEN2001
208 IFMID$(I$,J,1)=T$THENF=C:C=B:B=J:IFF>0THEN210
209 J=J-1:IFJ>0THEN208
210 S=L:E=L:IFJ=0ORF=0THEN2001
212 S$=MID$(I$,B+1,C-B-1):IFO=5THENE$=S$:E=1E5:GOTO214
213 E$=MID$(I$,C+1,F-C-1)
214 R=9:IFP<BTHENGOSUB1000:IFR>0THEN2001
215 IFR=0THENS=VAL(T$):E=S:IFP<BTHENGOSUB1000
216 IFP<BANDR=2THENGOSUB1000:IFR>1THEN2001
217 IFR=0THENE=VAL(T$)
218 IFP<>BTHEN2001
220 B=0:C=U:IFC=0THEN2006
221 IFC>0ANDL(C)<STHENC=F(C):GOTO221
222 P=1:IFC=0ORL(C)>ETHEN240
223 I$=L$(C):M=C:T$=S$:GOSUB1100:F=P:IFP>0THEN235
224 B=1:L$(C)=MID$(I$,1,P-K-1)+E$:IFI=0THEN230
225 F=P:GOSUB1100:IFP>0THEN230
226 L$(C)=L$(C)+MID$(I$,F,P-F-K)+E$
227 IFI>0THEN225
229 IFO=5THEN2100
230 L$(C)=L$(C)+MID$(I$,F,LEN(I$)-F+1)
231 IFO=5THEN2100
232 IFI=2THENPRINTL(C)TAB(10)L$(C)
235 C=F(C):GOTO222
240 O=0:IFB=0THEN2007
241 IFI<2THEN2100
242 GOTO25
250 GOTO200
300 GOTO50
350 IFU=0THEN2006
351 I=1:GOSUB1000:K=1:IFR<0THEN355
352 IFT$="-"THENI=-1
353 GOSUB1000:IFR<>0THEN2001
354 K=VAL(T$)
355 J=M:IFI<0THEN370
360 IFJ=0THENPRINT"BOTTOM OF DATA SET REACHED":GOTO2100
361 IFK=0THEN2100
362 J=F(J):K=K-1:IFJ>0THENM=J
363 GOTO360
370 IFJ=0THENPRINT"TOP OF DATA SET REACHED":GOTO2100
371 IFK=0THEN2100
372 J=B(J):K=K-1:IFJ>0THENM=J
373 GOTO370
400 IFU=0THEN2006
401 M=U:GOTO2100
450 F=U:IFU=0THEN2006
451 IFF(F)>0THENF=F(F):GOTO451
452 M=F:GOTO2100
500 J=U:K=1:M=U
501 IFJ=0THEN2100
502 L(J)=K:K=K+1:J=F(J):GOTO501
550 S=L:GOSUB1000:IFR<>0THEN2001
551 I=VAL(T$):GOSUB1000:IFR=2THENGOSUB1000
553 IFR=0THENS=I:I=VAL(T$):GOSUB1000:IFR=2THENGOSUB1000:IFR=2THEN2001
555 E=S:IFR=0THENE=I:I=VAL(T$):GOSUB1000:IFR>1THEN2001
556 IFT$<>"BY"THEN559
557 GOSUB1000:IFR<>0THEN2001
558 BY=VAL(T$):IFBY<1E-4ORBY>1E4THEN2004
559 K=0:IFE<STHEN2004
560 J=I:C=0:B=0:F=0:R=U
561 IFR=0THEN2006
562 IFL(R)<=JTHENC=R
563 IFL(R)<STHENR=F(R):GOTO561
564 IFL(R)<=ETHENB=R
565 IFR=0THEN570
566 IFL(R)<=JTHENC=R
567 IFL(R)<=ETHENK=K+1:F=R:R=F(R):GOTO565
568 IFR>0ANDL(R)<=JTHENC=R:R=F(R):GOTO568
570 IFB=0THEN2002
571 IFO=12THENR=B:K=1:GOTO110
572 R=F(C):IFC=0THENR=U
573 IFR=BTHENR=F(F)
574 M=F:IFR>0ANDL(R)<=I+K*BYTHEN2005
575 R=B
576 I=VAL(STR$(I+BY)):L(R)=I:IFR<>FTHENR=F(R):GOTO576
580 IFB(B)=CORF=CTHEN590
582 IFU=BTHENU=F(F)
583 IFF(C)>0THENB(F(C))=B
584 IFF(F)>0THENB(F(F))=B(B)
585 IFB(B)>0THENF(B(B))=F(F)
586 B(B)=C:IFC>0THENF(F)=F(C):F(C)=B:GOTO590
587 F(F)=U:B(U)=F:U=B
590 GOTO2100
600 GOTO550
650 C=U:IFU=0THEN2006
651 FORI=1TOQ:IFC>0THENPRINTL$(C):C=F(C):GOTO653
652 PRINT
653 NEXT:GOTO25
700 REM
710 CLOAD*L:CLOAD*F:FORI=1TON:T$=""
711 WAIT6,1,1:K=INP(7):IFK<255THENT$=T$+CHR$(K):GOTO711
712 L$(I)=T$:NEXT:U=L(0):UN=F(0):M=U
720 FORI=1TON:B(F(I))=I:NEXT:B(U)=0:B(UN)=0:F(0)=0:B(0)=0:L(0)=0:GOTO25
750 REM
760 L(0)=U:CSAVE*L:F(0)=UN:CSAVE*F:FORI=1TON:J=LEN(L$(I)):IFJ=0THEN780
```

set according to line numbers. To aid in this latter method, two pointers, B and F, are set up to point to the backward line and forward line, respectively, relative to the line number being inserted.

The example in Fig. 5 will help make this process clear. In this example, the data set is initially empty; we are going to insert a single new line with line number 1. The numbers to the left of the slashes in the examples are the original values prior to inserting; the numbers to the right of the slashes are the new values after the INSERT command is executed. Thus, initially the data set appears empty, exactly as in Fig. 3 with no lines in the used chain, and with the unused chain beginning at slot 1 and completely filling up all ten slots of the data set.

After the insert operation, the used chain has a single entry starting at slot 1, and the unused chain has nine entries starting at slot 2 and ending at slot 10. Remember that a zero in the F or B matrix terminates a forward or backward pointer chain, respectively. Also, recall that each of the F and B columns actually contains two chains, one consisting of used entries and one consisting of unused entries.

Finally, and most important, the entire operation of the INSERT command is portrayed by the appearance of slashes in the example—all of these operations are nothing more than pointer changes, and the relatively high speed of these operations, even though written in a high-level language, enables the editor to respond quickly.

In order to actually perform the pointer changes shown in the example, the INSERT command code must first locate the lines where the pointer changes are to take place. There are three variables, B, F and C, which appear at the bottom of each example, and are set up to point to these key lines (don't confuse the single variables B and F with the matrix variables B( ) and F( )).

As shown in Fig. 5, C points to the current slot, or the slot

Radio Shack's personal computer system? This ad just might make you a believer.

You can't beat the 4K system at $599

TRS-80 "Breakthru"
- TRS-80 microcomputer
- 12" video display
- Professional keyboard
- Power supply
- Cassette tape recorder
- 4K RAM, Level-I BASIC
- 232-page manual
- 2 game cassettes

...or the step-up 16K system at $899

TRS-80 "Sweet 16"
- Above, except includes 16K RAM

...or the fast 4K/printer system at $1198

TRS-80 "Educator"
- Above, except includes 4K RAM and screen printer

...or the Level-II 16K/printer/disk system at $2385

TRS-80 "Professional"
- Above, except includes 16K RAM, disk drive, expansion interface, and Level-II BASIC

So how are you gonna beat the system that does this much for this little? No way!

...The amazing new 32K/Level-II/2-disk/ line printer system at $3874

TRS-80 "Business"
- Above, except includes 32K RAM, line printer, and two disk drives

Get details and order now at Radio Shack stores and dealers in the USA, Canada, UK, Australia, Belgium, Holland, France, Japan. Write Radio Shack, Division of Tandy Corporation, Dept. C-046/B, 1400 One Tandy Center, Fort Worth, Texas 76102. Ask for Catalog TRS-80.

Radio Shack® R17

The biggest name in little computers ™

Prices May Vary at Individual Stores and Dealers

```
770 FORK=1TOJ:R=ASC(MID$(L$(I),K,1))
771 WAIT6,128,128
772 OUT7,R:FORP=1TO5:NEXTP,K
780 WAIT6,128,128
781 OUT7,255:NEXTI:GOTO25
800 INPUT"** DO YOU REALLY WANT TO KILL THE DATA SET (Y/N)";I$
801 IFLEFT$(I$,1)="Y"THENPRINTCHR$(26):GOTO10
802 GOTO25
850 REM
851 POKE2507,58:POKE2509,44
852 PRINT"NORMAL END":END
900 PRINT"UNUSED ="UN"  USED ="U"  B ="B"  F ="F"  C ="C:PRINT
901 PRINT" I"," F"," B"," L"
902 PRINT:FORI=1TOPW:PRINTI,F(I),B(I),L(I):NEXT:GOTO25
1000 R=-1:J=LEN(I$):IFP>JTHENT$="":RETURN
1001 FORP=PTOJ:IFMID$(I$,P,1)=B$THENNEXT:RETURN
1002 R=2:K=P:T$=MID$(I$,P,1):IFT$>=A$ANDT$<=Z$THEN1010
1003 IFT$>=X$ANDT$<=Y$ORT$=D$THEN1007
1004 P=P+1:RETURN
1007 FORP=PTOJ:T$=MID$(I$,P,1)
1008 IFT$>=X$ANDT$<=Y$ORT$=D$THENR=0:NEXT
1009 T$=MID$(I$,K,P-K):RETURN
1010 FORP=PTOJ:T$=MID$(I$,P,1)
1011 IFT$>=A$ANDT$<=Z$THENR=1:NEXT
1012 T$=MID$(I$,K,P-K):RETURN
1100 K=LEN(T$):FORJ=PTOLEN(I$):IFMID$(I$,J,K)<>T$THENNEXT:R=1:RETURN
1101 IFLEN(I$)=0THENI=0
1102 P=J+K:R=0:RETURN
2000 PRINT"NOT A COMMAND":GOTO25
2001 PRINT"SYNTAX ERROR":GOTO25
2002 PRINT"ILLEGAL LINE NUMBER":GOTO25
2003 PRINT"EDITOR BUFFER FULL":GOTO2100
2004 PRINT"PARAMETER VALUE ERROR":GOTO25
2005 PRINT"NOT ENOUGH SPACE TO INSERT":GOTO2100
2006 PRINT"THE DATA SET IS EMPTY":GOTO25
2007 PRINT"TEXT COULD NOT BE FOUND":GOTO2100
2100 IFM=0THEN2006
2102 PRINTL(M)TAB(10)L$(M):GOTO25
OK
```

where the new record will be inserted. B is a backward pointer which points to the line with the next lowest line number compared to the line number of the line being added. F is a forward pointer and points to the line with the next higher line number compared to the line number of the line being added. It is on these three lines that all action or pointer changes will occur.

The part of the insert algorithm that searches through the data set and sets up these three pointers is shown in Fig. 4 from point A to B. Once we are

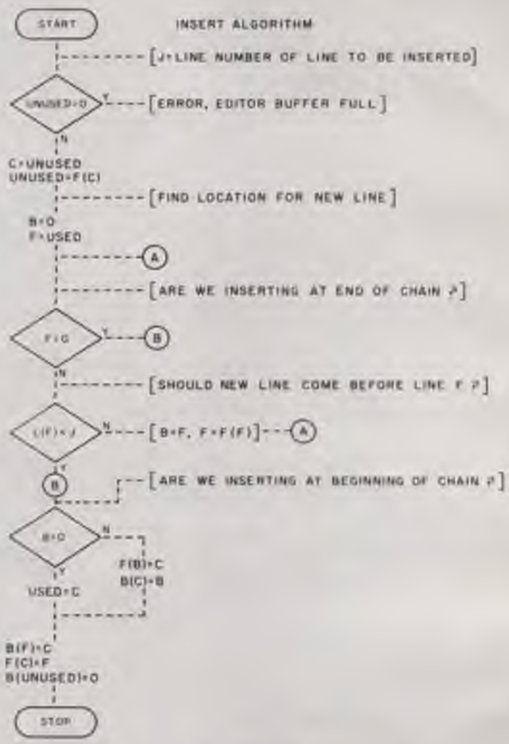


Fig. 4. Insert algorithm.

```
UNUSED:    1/2            INSERTING 1
USED:      0/1

I()     F()     B()     L()

1       2/0     0       [1]
2       3       1/0
3       4       2
4       5       3
5       6       4
6       7       5
7       8       6
8       9       7
9       10      8
10      0       9

B=0     F=0     C=1
```

Fig. 5.

```
UNUSED:    1/5            INSERTING 18
USED:      3

I()     F()     B()     L()

1       5/2     0/4     [18]
2       0       4/1     20
3       4       0       1
4       2/1     3       4
5       6       1
6       7       5
7       8       6
8       9       7
9       10      8
10      0       9

B=4     F=2     C=1
```

Fig. 6.

```
UNUSED:    4/5            INSERTING 4
USED:      3

I()     F()     B()     L()

1       2       3/4     17
2       0       1       20
3       1/4     0       1
4       5/1     0/3     [4]
5       6       4/0
6       7       5
7       8       6
8       9       7
9       10      8
10      0       9

B=3     F=1     C=4
```

Fig. 7.

at point B, the actual pointer changes can begin.

In performing the final part of the insert operation, four cases must be handled: (1) the new record, the only one in the data set; (2) the new record at the end of the data set; (3) the new record at the front of the data set; and (4) the new record within the middle of the data set. The insert algorithm must alter the data set pointers differently for each of these cases.

The illustrations in Figs. 5, 6 and 7 show how the algorithm reacts to different situations by varying the pointers themselves as well as the actual number of pointers that are changed.

DELETE. Fig. 8 shows the algorithm for deleting a line from the data set. The first step is to locate the slots in the data set where pointers will change. To do this, the program searches the used chain of the data set and sets up the variables B and F to key slots.

Fig. 9 illustrates this with an example. The variables B and F appear at the bottom of the illustrations. B and F are set up to point to the slot containing the line number and text to be deleted.

After the slots where pointer changes are to occur are located, the final operation of the DELETE command is to change pointers on these lines to maintain the used chain in numerically ascending order of line numbers.

There are three cases the algorithm must consider: (1) the record to be deleted is at the front of the chain; (2) the record to be deleted is at the back of the chain; (3) the record to be deleted is within the middle of the *chain.* The examples in Figs. 9 and 10 illustrate several

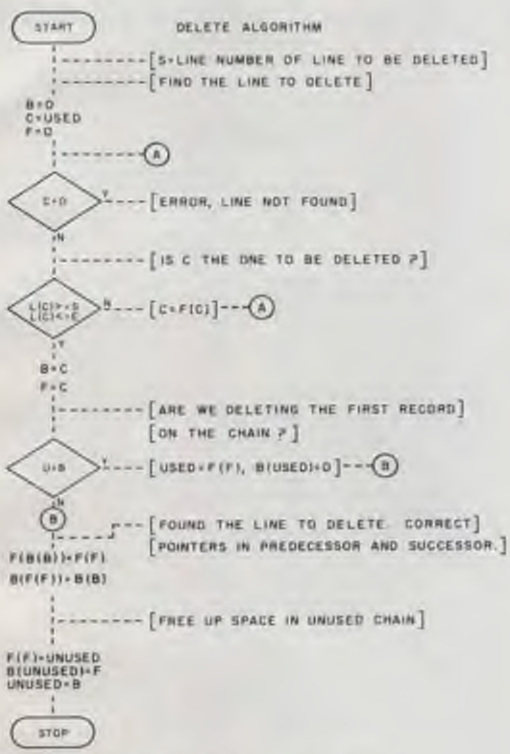


Fig. 8. Delete algorithm.

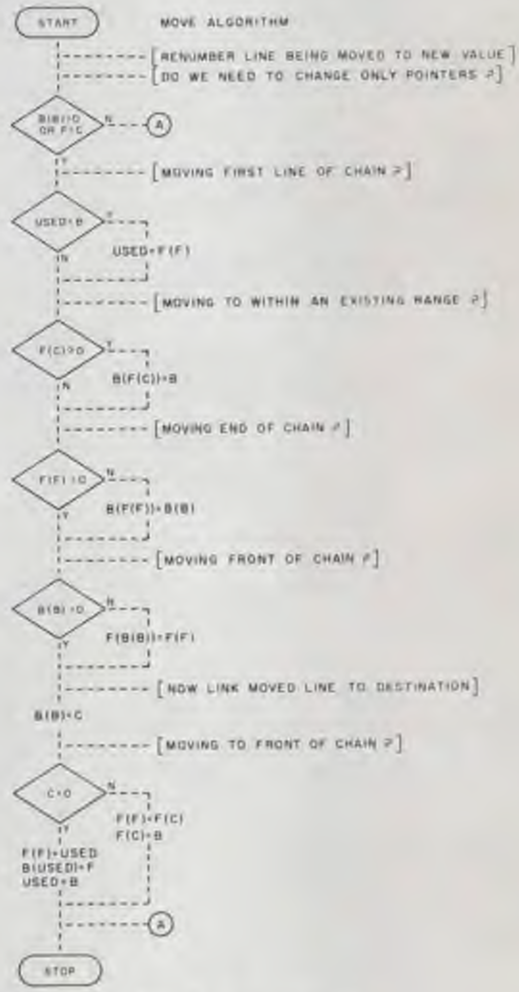


Fig. 11. Move algorithm.

UNUSED: 2/1 DELETING 1
USED: 1/0

I()	F()	B()	L()
1	0/2	0	[1]
2	3	0/1	
3	4	2	
4	5	3	
5	6	4	
6	7	5	
7	8	6	
8	9	7	
9	10	8	
10	0	9	

B=1 F=1

Fig. 9.

UNUSED: 5/1 DELETING 17
USED: 3

I()	F()	B()	L()
1	2/5	4/0	[17]
2	0	1/4	20
3	4	0	1
4	1/2	3	4
5	6	0/1	
6	7	5	
7	8	6	
8	9	7	
9	10	8	
10	0	9	

B=1 F=1

Fig. 10.

of these cases.

Note that the record is really not physically deleted, but rather, pointers are changed so that it is transferred from the used chain to the unused chain and thus appears deleted. This is a design feature of the editor that greatly saves time.

MOVE. The MOVE command algorithm is shown in Fig. 11, with supporting examples in Figs. 12, 13 and 14. Once again, the first part of the algorithm involves identifying the slots where pointers must be changed. The three variables B, F and C are set up to perform this function; these variables appear at the bottom of each example.

B and F both point to the slot containing the line to be moved. C points to the slot whose line number is immediately before the destination line number of the line being moved. Or, in other words, after the line is moved, C will point to the slot containing the immediately preceding line.

After B, F and C have been set up, the final portion of the move algorithm is executed. This part changes pointers in the data set in order to implement the move.

There are several important cases that the algorithm must handle: (1) a line is being moved from the front of the data set; (2) a line is being moved from the back of the data set; (3) a line is being moved from within

```
UNUSED:    5              MOVE 1 TO AFTER 20
USED:       3/4

I()       F()       B()       L()

1         2         4         17
2         0/3       1         20
3         4/0       0/2       [1]
4         1         3/0       4
5         6         0
6         7         5
7         8         6
8         9         7
9         10        8
10        0         9

B=1     F=1     C=2
```

Fig. 12.

```
UNUSED:    4              MOVE 2 TO BEFORE 1
USED:       1/2

I()       F()       B()       L()

1         2/3       0/2       1
2         3/1       1/0       [2]
3         0         2/1       3
4         5         0
5         6         4
6         7         5
7         8         6
8         9         7
9         10        8
10        0         9

B=2     F=2     C=0
```

Fig. 13.

```
UNUSED:    5              MOVE 17 TO AFTER 1
USED:       3

I()       F()       B()       L()

1         2/4       4/3       [17]
2         0         1/4       20
3         4/1       0         1
4         1/2       3/1       4
5         6         0
6         7         5
7         8         6
0         9         7
9         10        8
10        0         9

B=1     F=1     C=3
```

Fig. 14.

N	NUMBER OF LINES IN EDITOR BUFFER (CURRENTLY 60)
B(N)	BACKWARD POINTER ARRAY
F(N)	FORWARD POINTER ARRAY
L(N)	LINE NUMBER ARRAY
L$(N)	LINE TEXT ARRAY
A$	"A"
Z$	"Z"
X$	"0"
Y$	"9"
D$	"."
B$	" "
C$	"DIPCFONTBRMEWLSKQX" COMMAND STRING CONTAINING FIRST LETTER OF EACH COMMAND.
O	OPERATION CODE 1 = DELETE, 2 = INSERT, 3 = PRINT, 4 = CHANGE 5 = FIND, 6 = OVERLAY, 7 = NEXT, 8 = TOP 9 = BOTTOM, 10 = RENUMBER, 11 = MOVE, 12 = EXCERPT 13 = WRITE, 14 = LOAD, 15 = SAVE, 16 = KILL 17 = QUIT, 18 = XRAY
I$	INPUT STRING FROM USER'S TERMINAL
T$	TEMPORARY STRING
S$	OLD TEXT
E$	NEW TEXT
B	BACKWARD POINTER
C	CURRENT POINTER
F	FORWARD POINTER
L	CURRENT LINE NUMBER
M	CURRENT LINE INDEX
P	CHARACTER POINTER
U	USED CHAIN POINTER
UN	UNUSED CHAIN POINTER
BY	LINE NUMBER INCREMENT VALUE
I	TEMPORARY VARIABLE
J	TEMPORARY VARIABLE
K	TEMPORARY VARIABLE
R	RETURN CODE FROM TEXT PROCESSING SUBROUTINE

Fig. 15. Program variables.

[LINE NUMBER]	[FUNCTION]
1-24	INITIALIZATION
25-49	COMMAND PROCESSOR
50-99	DELETE
100-149	INSERT
150-199	PRINT
200-249	CHANGE
250-299	FIND
300-349	OVERLAY
350-399	NEXT
400-449	TOP
450-499	BOTTOM
500-549	RENUMBER
550-599	MOVE
600-649	EXCERPT
650-699	WRITE
700-749	LOAD
750-799	SAVE
800-849	KILL
850-899	QUIT
900-949	XRAY
1000-1399	EXTRACT NEXT PARAMETER
2000-2099	PRINT ERROR MESSAGES
2100-2199	PRINT CURRENT LINE

Fig. 16. Program index.

the data set; (4) a line is being moved to the front of the data set; (5) a line is being moved to the back of the data set; (6) a line is being moved to within the data set.

Note that as shown in the examples, no data is physically moved. Rather, all pointers are rearranged to relink the data set so it appears to have been moved. Again, this design feature saves time.

Conclusion

Version 5.0 of the editor adds a new philosophy to DELETE, OVERLAY and MOVE commands. Rather than operate on one line at a time as shown in the examples, these commands now operate on an entire range of lines at once. Thus, while it used to take roughly five times as long to delete five lines as it did for one line, it now takes no additional time to DELETE or OVERLAY a complete range of lines.

The MOVE command operates similarly, but is not as fast since it still must physically change all line numbers in the range of lines being moved. Also, the MOVE command has a nice feature that first checks if the entire move can take place successfully. If it determines that it can, it will go ahead and execute it; otherwise, it will print an error.

Many commercial editors operate on a line-at-a-time basis and can leave you stranded halfway through a move if the current line being moved won't fit at its new destination. Interestingly enough, the algorithms to operate on an entire range are nearly the same as those for the single line case; the main difference is that the B and F single variables are set up differently to start with.■

NOT A COMMAND
: The command entered is not one of the legal recognized commands available. See Fig. 1 for the list of possible commands.

TEXT COULD NOT BE FOUND
: The text being searched for could not be found within the range of lines being operated on.

BOTTOM OF DATA SET REACHED
: The user attempted to move beyond the current lower boundary of the data set.

TOP OF DATA REACHED
: The user attempted to move beyond the current upper limit of the data set.

THE DATA SET IS EMPTY
: The user requested a command but there is no data to act upon.

SYNTAX ERROR
: Error in command format, options, or delimiters which results in ambiguous interpretation by the editor.

ILLEGAL LINE NUMBER
: Line specified which does not exist or is out of range.

EDITOR BUFFER FULL
: The editor buffer is filled to capacity. The program can be changed to extend this limit.

PARAMETER VALUE ERROR
: By value is < .0001 or > 10,000 or last line of range is less than first line of range.

NOT ENOUGH SPACE TO INSERT
: Attempt to insert or move on top of existing lines. This usually means the BY value is not low enough to allow the full range of lines to be inserted or moved. The last line printed is the last line moved or inserted. If the BY value is already at its lowest allowed value (.0001) and there is still no room, the RENUMBER command can be invoked to clean up the data set and provide additional room.

Fig. 17. Output messages.

1) For those who want to label the pages of text they are saving and loading, the SAVE and LOAD commands can be changed to SAVE<NAME> and LOAD<NAME>, where <NAME> is a single character file name. To add this tape file header feature, insert the following lines:

```
701  GOSUB1000:IFR<0THEN710
702  R=ASC(LEFT$(T$,1))
703  WAIT6,1,1:IFINP(7)<>RTHEN703
751  GOSUB1000:IFR<0THEN760
752  WAIT6,128,128:OUT7,ASC(LEFT$(T$,1))
```

2) For Mits 8K BASIC 4.0, add the following lines to the code to allow commas & colons to appear in text:

```
6    POKE2507,34:POKE2509,34
851  POKE2507,58:POKE2509,44
```

By doing this, BASIC will be altered upon editor initialization, and restored to normal when the user leaves the editor via the QUIT command.

Fig. 18. Special options.

Bill Welborn
2221 W. Franklin St.
Evansville IN 47712

At Last: A Client Timekeeping System

Many different businesses have to accurately keep track of the time expended, or spent with a client . . . for billing purposes. What better way to put the computer to work?

This is a fantastic article . . . for several reasons. First, if you want to see someone get really excited about it, try taking it to your lawyer and see what he has to say. A lawyer once said that having a computer only for keeping track of client time would be worthwhile. No telling how true that was, but the value he placed on such a program was certainly impressive. Then along comes Bill Welborn, attorney-at-law, who has generated such a program! Aside from the value of the program itself (and it applies to businesses other than law firms), there are several "hidden" features that make the article worthwhile reading for everyone. Bill discusses some conversion techniques going from IBM 370 BASIC to some of the popular microsystem BASICs . . . and he has some interesting comments on dealing with customers and the importance of backup data files.—Eds.

Do you own a business where clients or customers are billed by the hour? . . . know anyone who does? . . . want to set up a computer-billing service for such a business? If you can answer any of these questions in the affirmative, these programs are for you.

Such businesses include law offices, accounting firms, surveying companies, consulting firms and others in which a client comes in, describes his problem, and the firm charges an hourly rate for trying to work that problem through to a solution. The rate might run anywhere from ten to one hundred dollars per hour, depending on the type of firm and the nature of the problem.

Although IBM, DEC and other hardware manufacturers have excellent program packages for hourly billing (which work only on their $30,000 wonders, of course), they are about the only companies offering such software for the above businesses. For example, although there are five computer-billing services in my area (a city of over 175,000), none of them offers a billing service tailored to the needs of the professional office. Their programs are for department-store and other retail operations. I have presented here two programs that just might give you a start on that market in your town, even if you can't use

```
HOURS

 10 REM     INPUTS DATA FOR HOURLY CHARGES.
 15 REM
 20 PRINT "NOTE: FOR EXPENSES, ENTER AN 'E' INSTEAD OF DATE, THEN":
 30 PRINT "THE DATE-SERVICE AS ONE ITEM, A ZERO AND, FINALLY, THE AMOUNT."
 35 REM
 40 REM     NAME THE FILE AND INITIALIZE THE RECORD POINTER (C)
 50 FILES HOURS1
 60 LET C=0
 70 REM     TELL THE CPU WHERE TO GO ON FINDING A BLANK RECORD TO WRITE ON
 80 IF END OF 1 THEN 130
 90 REM     FIND BLANK RECORD
 95 REM
100 LET C=C+1
110 READ #1,C;A$
120 GOTO 100
130 PRINT C: "CLIENT NAME,PROJECT":
140 REM     C (RECORD NUMBER) SHOWS WHAT RECORD YOU ARE ON.
150 REM     WHEN INSTRUCTIONS ABOVE PRINT, A BLANK REC. HAS BEEN FOUND
160 REM     AND IS WAITING FOR DATA INPUT.
170 INPUT A$,B$
180 PRINT "DATE,SERVICE,BY,HRS":
190 REM     ABOVE PROMPTS THE OPERATOR.  NOW, ALLOW ENTRY OF DATA.
200 INPUT M$,C$,P,D
205 REM
210 REM     GIVE OPERATOR A CHANCE TO CORRECT ERRORS AND THEN WRITE TO FILE
220 PRINT "ALL CORRECT":
230 INPUT E$
240 IF E$="N" THEN 130
250 WRITE 1,C:A$,B$,M$,C$,P,D;END
260 PRINT "ANOTHER ENTRY":
270 INPUT F$
280 IF F$="Y" THEN 90
290 REM     IF THERE IS MORE DATA, WE LOOP AND GET NEW RECORD. IF NOT, END IT.
300 END
```

Fig. 1. HOURS. The data input program written in IBM 370 BASIC, Version 3.1. See text for details.

HOURS—variables:

Name	Description
C	Record counter (pointer)
A$	Client names
B$	Client project name
M$	Date field (Also used as key when it contains an E)
C$	Service rendered
P	Partner code—who did work
D	Hours used for service (or dollar amount in an expense record)

HRSCAL—variables:

SAME AS ABOVE, EXCEPT:

Name	Description
U$	Date of billing
Z	Rate per hour
Z1	Product of rate times hours
Z2	Z1 rounded to cents (#####.##)
D1	Total of hours
D2	Total of expense record dollar amounts
D3	Grand total of Z2 + D2 (amount of bill)
M, K	Record counters (pointers)
X$,Z$	Same as A$,B$, except input from terminal

Fig. 2. List of variables used in both programs. Most micro-based BASICs permit more than one letter + number variable names; rename the variables for clarity if you wish.

them yourself.

Remember, too, that to the extent you use your computer for a business purpose, you can probably deduct a portion of the expense for upkeep (as well as depreciate a portion of the cost of your system) when April 15 rolls around. See your accountant or tax lawyer for details (and hope he is right!).

Why the Programs?

I am a member of a three-man law office. Several years ago, I designed (but did not program—it's in IBM 370 assembly language!) a fairly extensive billing system for our office. This system is still running on a line to the IBM 370, but soon we will have our own micro doing the work. Anyway, the main system does not have provisions for hourly billing of clients; we generally just don't work that way. Usually, when a client has a case for us, we quote him a fee out front and that's that—if we go over, we just have to eat the loss.

However, after the system was in operation, I found out that there were enough exceptions to our usual way of doing business that it was rapidly becoming a real drag to do the hourly-rate billing manually. So, I asked the people who did the original programming to quote me a price. Wow! . . . it would astound you. They had apparently undercharged a great deal on the main package and figured it was time to make up the loss. So, forget that!

Then, after a while, I asked the computer center where the IBM monster is located how much more a month it would cost us to add BASIC facilities (including three or four data files) to our system. Believe it or not, the answer was $35, plus $50 per hour of CPU time over one-half hour. Man, I was in! Not only could I have our hourly billing done by computer, but I could also have a lot of fun playing around with all the *Kilobaud* programs that were just then being published.

I ordered the BASIC hookup, enrolled in a computer-science course at one of the local universities, and got to work. (By the way, we have yet to use that first half hour of CPU time, despite extensive use of the BASIC system for much more than is described here.) As a novice at programming, I took a while to come up with the right concept for hourly billing—and a little while longer to get the programs to work as I wanted them to. But, I did learn a lot . . . I still am. And, the programs *do* work.

Some Road Maps

We're going to go through the programs in detail together. Where it seems appropriate, I'll comment on features not found on most micro-BASICs and what to do about the problems thus presented (later, I will present a separate section on this same subject).

Then, and throughout, I'm going to beat you to the draw and make some critical comments about the programs: They *are* basic, as well as BASIC. I will have some comments and suggestions about making the programs better as well as modifications for faster operation and different conceptions of the problem to be solved.

Finally, if you're still with me at that point, we will explore some ways that the programs can be made to work with tape instead of disk. Get your computer's BASIC manual and a beer (or something) and let's get started.

General Information

There are two main programs: HOURS and HRSCAL. The first permits entry onto a file of time spent for particular clients, and the second program prints the bills. Both programs assume that you are working with files on disk that: (a) have been allocated in the system; (b) are random access; (c) start with record #1; (d) contain 128 bytes per record; and (e) have a maximum of 256 records per file. If these assumptions are not valid for your system, please don't give up just yet; the programs (as listed) are not really random access and should work with tape, as we will discover later.

Hours

The listing is in Fig. 1; the list of variables is shown in Fig. 2. In the following text, numbers in parentheses are statement numbers. For now, forget statement 20, alright?

First, open the file for access (read and write). This is the statement FILES HOURS1 (50). HOURS1 is the name of the file, and the statement works in the IBM implementation of BASIC, but probably won't in yours. You'll just have to look in your manual and see what you need to do. For example, in the excellent BASIC implementation by Alpha-Micro Technology for their 16-bit microcomputer, the statement would be:

```
50 OPEN #1, "HOURS1",RANDOM,64,C
```

In most all disk systems, after the file has been opened it

```
260 REM   IF MORE FOR SAME CLIENT & PROJECT, STEP RECORD
270 REM   COUNTER AND PROCEED--NO SEARCH FOR BLANK RECORD.
280 PRINT "ANOTHER ENTRY FOR SAME CLIENT AND PROJECT":
290 INPUT F$
295 IF F$="N" THEN  340
300 LET C=C+1
310 GOTO 180
320 REM   IF THERE IS NO MORE DATA FOR SAME CLIENT, SEE IF
330 REM   THERE IS MORE FOR ANOTHER: IF SO, LOOP.
340 PRINT "MORE FOR DIFFERENT CLIENT OR PROJECT":
350 INPUT F$
360 IF F$="Y" THEN 100
370 REM   IF NO FURTHER DATA, ENDIT
380 END
```

Fig. 3. Additions/substitutions to HOURS. Use is explained in text. Please observe cautions outlined there.

```
IF C>255 THEN PRINT "FILE FULL": GOTO 300
```

Example 1.

```
HOURS     3/8/78    WED 14:21

NOTE: FOR EXPENSES, ENTER AN E INSTEAD OF THE DATE, THEN
THE DATE-SERVICE AS ONE ITEM, 'BY' AND, FINALLY, THE AMOUNT

 1 CLIENT NAME,PROJECT? JOHN SMITH, DIVORCE
DATE,SERVICE,BY,HRS? 11/1/77,INITIAL CONFERENCE,3,2.5
ALL CORRECT? Y
 2 CLIENT NAME,PROJECT? PETE JONES,TRUSTEE
DATE,SERVICE,BY,HRS? 1/12/77,TELCON JUDGE RE APPOINTMENT,2,.5
ALL CORRECT? Y
ANOTHER ENTRY? Y
 3 CLIENT NAME,PROJECT? JOHN SMITH,PETERS CONTRACT
DATE,SERVICE,BY,HRS? 12/1/77,REVIEW OLD CONTRACT,1,3.25
ALL CORRECT? Y
ANOTHER ENTRY? Y
 4 CLIENT NAME,PROJECT? JOHN SMITH,DIVORCE
DATE,SERVICE,BY,HRS? E,11/2/77  FILING FEE,0,28
ALL CORRECT? Y
ANOTHER ENTRY? Y
 5 CLIENT NAME,PROJECT? PETE JONES,TRUSETT
DATE,SERVICE,BY,HRS? 1/15/78,COURT HEARING,2,4.5
ALL CORRECT? N
 5 CLIENT NAME,PROJECT? PETE JONES,TRUSTEE
DATE,SERVICE,BY,HRS? 1/15/78,COURT HEARING,2,4.5
ALL CORRECT? Y
ANOTHER ENTRY? Y
 6 CLIENT NAME,PROJECT? JOHN SMITH,DIVORCE
DATE,SERVICE,BY,HRS? 11/6/77,FINAL HEARING,2,1.5
ALL CORRECT Y
 7 CLIENT NAME,PROJECT? JOHN SMITH,PETERS CONTRACT
DATE,SERVICE,BY,HRS? E,12/3/76   L.D.PHONECALL:N.Y.,2,5.67
ALL CORRECT? Y
ANOTHER ENTRY? N
```

Underscored items were input by operator.

Fig. 4. Actual entry using HOURS. In this case, file was empty, so entries started with record one. Note the error made at record #5 and the subsequent correction. If this had not been corrected, the record would have been bypassed by HRSCAL. (See text discussion.) Compare the entries to the bills printed by HRSCAL (Fig. 6).

is referenced by number. Since there is only one file in the HOURS program, it is referenced by the number 1. Frequently, a pound sign (#) is used before the file number to indicate that it is a disk file.

After the file has been opened, we have to find a blank record (or a closed one) on which to write new data. This is the purpose behind statements 80 through 120; these deserve a lot of explanation. I'll try to be brief, without being too cryptic.

The variable C is a record pointer that starts at 1 and is incremented by 1 each time through the loop formed by statements 100 through 120. Now, this loop looks as though it will go on endlessly reading record after record—never printing anything and never stopping. Ha! The key to this particular madness is statement 80. In most large-computer BASICs, any blank record anywhere in the file is full of ASCII nulls. When the computer reads a null record, it will look for an IF END THEN . . . statement and will go where that statement directs it. If there is no IF END statement, then you will get a program termination and a nasty message.

So, when the computer comes (in the course of the loop) to a null record and tries to read it, the program execution jumps to statement 130 because of the IF END statement. The absence of an IF END statement (function) in most microcomputer BASIC implementations is problem number one. We will solve that one together later on. Take it on faith for now, OK?

At statement 130, the record number found is printed, together with an operator prompt: "CLIENT NAME, PROJECT". Then (170) the operator inputs the client's name and the project for that client. Note, please, that there are two string variables here. These are *not* for last name and first name. A$ is for *both* names and B$ is for the particular project (case) name for that client. In the print program, bills are separated both by name and project. One client, for example, might get five bills, if he has five cases with the office. This is done for the client's bookkeeping, so he can allocate his costs properly.

After this data has been input, another prompt gives the operator the cue to enter date, service rendered, by whom it was rendered and the number of hours (to the nearest tenth) it took to do the work.

There are a couple of things to note here. First, you may wonder why the date is a string and not a numeric variable. OK, *now* look back at statement 20. If the date-field has an E in it, the item is an expense item and we are going to do something different with it in the print program. Any business using these programs will have expenses that are to be charged to the client: phone calls, copy costs, travel and so on. The above is a way to get those expenses in. There are other ways, of course, . . . better ways, maybe, especially from the operator (human-engineering) standpoint. But, this way

Fig. 5. HRSCAL listing. See text for details.

```
 10 REM         THIS PGM. CALCULATES TOTAL HOURLY CHARGES
 20 REM         FOR A NAMED CLIENT AND PROJECT.
 30 LET U$=DATE
 40 PRINT SKP(2)
 50 REM     NAME THE SOURCE FILE AND THE WORK FILE.  CLEAR THE WORK FILE.
 60 REM     ZERO EVERYTHING AND TELL THE CPU WHAT TO DO WHEN IT FINDS ALL
 70 REM     THE RECORDS YOU WANT IT TO FIND.
 75 REM
 80 FILES HOURS1,TRUST2
 90 ERASE 2
100 LET Z,Z1,Z2,D1,D2,D3,C,M,K=0
110 IF END OF 1 THEN 340
120 PRINT "CL.NAME,PROJECT":
130 REM    ENTER THE CLIENT'S NAME,PROJECT,AND HOURLY CHARGES
140 INPUT X$,Z$
150 PRINT "RATE":
160 INPUT Z
170 REM     SEARCH THE SOURCE FILE FOR THE NAMED CLIENT
180 LET C=C+1
190 READ #1,C;A$,B$,M$,C$,P,D
200 IF A$=X$ THEN 220
210 GOTO 180
220 IF B$=Z$ THEN 260
230 GOTO 180
240 REM     IF THE FILE NAMES MATCH BUT THERE IS AN 'E' IN THE DATE FIELD,
250 REM     GO ADD THE EXPENSE AMOUNT BEFORE WRITING TO THE WORKFILE.
260 IF M$="E" GOSUB 970
270 REM     ADVANCE RECORD POINTER IN WORKFILE AND WRITE THE DATA.
280 LET M=M+1
290 WRITE 2,M;A$,B$,M$,C$,P,D;END
300 REM     LOOK FOR THE NEXT MATCH IN SOURCE FILE--LOOP TILL FOUND
310 GOTO 180
320 REM     WHEN ALL MATCHES HAVE BEEN FOUND, THE WORK FILE IS COMPLETE:
330 REM     AND WE START PRINTING OUT THE BILLING FOR THIS CL-PROJ.
340 PRINT SKP(2)
350 REM         AFTER PRINTING THE ABOVE TWO BLANK LINES, FORMAT THE BILL,
360 REM         PRINT HEADINGS AND TELL THE CPU WHAT TO DO WHEN IT RUNS OUT.
370 PRINT SPA(10):"************************************************************"
380 PRINT SPA(5):"ACCOUNT OF ":X$:"--":Z$";" AS OF ":U$
390 PRINT
400 PRINT SPA(2):"DATE":TAB(12):"SERVICE RENDERED":TAB(50):"BY":
410 PRINT TAB(58):"HOURS"
420 IMAGE "###.##"
430 IF END OF 2 THEN 550
```

```
440 REM        START THE WORKFILE RECORD POINTER AND GO----
450 LET K=K+1
460 READ #2,K;A$,B$,M$,C$,P,D
470 REM     HOLD UP FOR NOW IF AN EXPENSE ITEM FOUND--WILL PRINT LATER.
480 IF M$="E" THEN 450
490 REM     OTHERWISE, PRINT THE DATA(NOT NAMES) KEEPING TRACK OF TOT.HOURS
500 PRINT M$:TAB(10):C$:TAB(50):P:TAB(57):
510 PRINT USING 420,D
520 LET D1=D+1
530 GOTO 450
540 REM     NEXT STATEMENTS EXECUTE ONLY AFTER FIRST END OF FILE#2 REACHED
550 PRINT TAB(57):"--------"
560 PRINT TAB(40):"TOTAL HOURS":TAB(57):
570 PRINT USING 420,D1
580 REM     MULTIPLY RATE TIMES HOURS, ROUND TO CENTS AND PRINT IT
590 PRINT
600 IMAGE "#####.##"
610 LET Z1=Z*D1
620 LET Z2=INT(Z1*100+.5)/100
630 PRINT TAB(40):"TOTAL FEE IS":TAB(55):"$":
640 PRINT USING 600,Z2
650 PRINT
660 REM     NOW, GO BACK AND GET THOSE EXPENSE ITEMS; PRINT HEADINGS AND DATA
670 IF END OF 2 THEN 770
680 PRINT "LIST OF EXPENSES:"
690 PRINT "   DATE":TAB(12):"FOR":TAB(30):"BY":TAB(40):"AMOUNT"
700 LET K=0
710 LET K=K+1
720 READ #2,K;A$,B$,M$,C$,P,D
730 IF M$<>"E" THEN 710
740 PRINT C$:TAB(30):P:TAB(38):"$":
750 PRINT USING 600,D
760 GOTO 710
770 PRINT
780 REM     ADD ALL EXPENSES TO TOTAL HOURLY CHARGE AND PRINT TOTALS
790 LET D3=Z2+D2
800 PRINT TAB(40):"EXPENSES TOTAL":TAB(55):"$":
810 PRINT USING 600,D2
820 PRINT TAB(57):"--------"
830 PRINT TAB(32):"TOTAL THIS STATEMENT":TAB(55):"$":
840 PRINT USING 600,D3
850 PRINT SKP(2):TAB(20):"T H A N K   Y O U"
860 PRINT SKP(3)
870 PRINT SPA(10):"**********************************************************"
880 PRINT SKP(3)
890 REM     THE FOLLOWING GIVES YOU READING OF HOW MANY RECORDS WERE READ IN1.
900 PRINT "RECORDS READ, FILE1....":C-1
910 REM     IF THERE IS ANOTHER BILL TO PRINT, START OVER: OTHERWISE ENDIT.
920 PRINT "ANOTHER":
930 INPUT M$
940 IF M$="Y" THEN 30
950 GOTO 990
960 REM     FOLLOWS SUBROUTINE TO ADD UP EXPENSE AMOUNTS
970 LET D2=D+D2
980 RETURN
985 REM
990 END
```

is simple, and it works. Experience shows that the operators in our office, at least, are not even slightly confused by this (why, I don't know; it confuses me whenever I try to input this program).

Note, too, that the "BY" field is numeric—we use a number code to indicate who did the work. If initials seem better for you, then just change the variable P to P$ in all the programs.

It is a good idea to give the operator a chance to correct an entry if a goof is made (but not caught) before return is hit. This is done in statements 220 through 240. If all is OK, the operator inputs a "Y" and the data is written to the file. Then, we ask if there is more data. If so, the search for a blank record is done again and the program repeats. If not, the program ends.

Gaze, if you will, at Fig. 3. This code is actually what we use. It permits stepping the record counter by one and eliminating the entry of client name and project identification if there is another entry for the same client and project.

This is very handy, but I have left it out of the main listing for this reason: If you have erased a record (say, record #47) by filling it with nulls—either manually or by virtue of the computer's ERASE statement—the next record might be full of another client's information. When you search for the first blank record (for the first entry), you will, of course, find record #47. Then, the code in Fig. 3 would advance you to record #48, and would then write new data on top of what was already there, destroying the old data.

So, use the code in Fig. 3 only with this restriction in mind. Some later discussion might help here, but this is already complicated enough without further clutter at this point.

A sample of the entry you will get with HOURS is shown in Fig. 4. Notice, particularly, the incorrect entry and the correction thereof.

Before I go on: an aside. You may have wondered why the record pointer, C, is printed out before every entry. There are two reasons. First, I wanted to have a record of every entry on the hard-copy printer we use so that I have something to look back to for troubleshooting. This is valid, I think.

Second, however, when the operator sees 255 come up, he knows that the file is going to be full with the next entry and must terminate operations and call me. This is not valid—it is crude. In *your* version of this program, I suggest putting in a statement right after (100). For example, see Example 1. In other words, jump automatically to the end of program on getting a full file and after printing an error message. My operators are used to this, plus we have yet to fill a file before cleaning it out with billing—but why not be a little more elegant?

HRSCAL

The listing of the printout program is shown in Fig. 5. HRSCAL gets a bit more complex than HOURS. The entry program is very simple; HRSCAL is a little more difficult. It searches the file for a match between the client and project names input from the terminal and the client and project names in a record. If a match is found, the record is written to another file (TRUST2) that is used as a workfile. When the end-of-file marker is reached, the bill is formatted and then written from the information in the workfile. That's it, generally. Now, to specifics.

Statement 30 uses a built-in function of IBM BASIC. If you don't have it, use an input statement for the date so your client will know when the bill was prepared.

Both files are opened for access, the second one (workfile) is totally cleared of data and all variables are zeroed (80-100).

Now, we hit that IF END bit again (110). Here, it is used to signal the program to go from the search mode to the printing mode. We'll get into this in some detail in the IF END . . . THEN . . . section of the article. Don't let the absence of the IF END in your BASIC turn you off; we'll make the programs work anyway.

The operator now inputs the client's name and project (140). It is obvious to you: (a) that a comparison is about to be made of maybe 40 bytes or so of alphanumeric data and (b) that this is not only inefficient but that an operator error on entry of only one letter could cause records to be lost forever. All true.

```
HRSCAL  06/05/78  MON 16:16


CL.NAME, PROJECT
? JOHN SMITH,DIVORCE
RATE? 40

            ***********************************************************
        ACCOUNT OF JOHN SMITH -- DIVORCE AS OF 06/05/78

  DATE       SERVICE RENDERED                              BY       HOURS
11/1/77   INITIAN CONFERENCE                               3         2.50
11/6/77   FINAL HEARING                                    2         1.50
                                                                   --------
                                                 TOTAL HOURS        4.00

                                                 TOTAL FEE IS    $  160.00

LIST OF EXPENSES:
   DATE      FOR                    BY          AMOUNT
11/2/77  FILING FEE                 0       $    28.00

                                                 EXPENSES TOTAL $   28.00
                                                                   --------
                                          TOTAL THIS STATEMENT   $  188.00

                           T H A N K   Y O U

            ***********************************************************


RECORDS READ  7
ANOTHER? Y
CL.NAME, PROJECT
? JOHN SMITH,PETERS CONTRACT
RATE? 50

            ***********************************************************
        ACCOUNT OF JOHN SMITH -- PETERS CONTRACT AS OF 06/05/78

  DATE       SERVICE RENDERED                              BY       HOURS
12/1/77   REVIEW OLD CONTRACT                              1         3.25
                                                                   --------
                                                 TOTAL HOURS        3.25

                                                 TOTAL FEE IS    $  162.50

LIST OF EXPENSES:
   DATE      FOR                    BY          AMOUNT
12/3/76  L.D.PHONECALL:N.Y.         2       $     5.67

                                                 EXPENSES TOTAL $    5.67
                                                                   --------
                                          TOTAL THIS STATEMENT   $  168.17

                           T H A N K   Y O U

            ***********************************************************
```

Our firm uses this method because client numbers are used on the main system and there could be a lot of confusion between files. I try to eliminate errors by providing the operator with a list of client names and projects that will be used *every* time an hourly client's name and project are input to the file.

If you want a faster search and less room for error, by all means use client numbers. The change will only require an additional variable in the record (in HOURS and HRSCAL) and a modification of the search routine in HRSCAL to look for a client number input at statement 140. Of course, each project should be logged under a different number in order to keep the feature of different bills for different projects.

To give you some idea if this is really worth changing the programs, I can tell you that the maximum search time using name and project comparison in alpha on our system is .2 seconds. I really doubt whether it would exceed five seconds in any micro-BASIC disk. And, remember, this is a search of all records to find the few for one client. So, I suggest you analyze the trade-off of CPU time vs programming effort and operator error possibility in *your* system and make your own decision. Printer speed is certainly a factor!

The actual search read and compare is at statements 180-230. The search and compare overhead is cut considerably by the multiple comparisons (200-220). That is, the name is compared first (maybe 20 bytes) and if that doesn't match, the program loops and doesn't bother about comparing the project ID.

At 260, the program looks at the date-field from the data just read from disk. If this is an E, remember, it indicates an expense item. Here is where we keep a cumulative total of the amount, to be used later.

Now, the workfile record pointer (M) is advanced one step and the client's data is written into the workfile (280-290). Then, we loop back to file #1 and look for another match (310).

When the end-of-file #1 is reached, the IF END statement at 110 sends the program to execute statement 340. This is where the printer formats and prints the bill. This part of the program is pretty self-explanatory. A bill header is printed and then items are read from each workfile record and printed according to the format of statement 500. At the same time, we keep a total of the hours (520). Expense records are bypassed for the moment (480).

You will note that another IF END statement crept in at statement 390. These are handy little things! The second one takes control and the first one is lost forever. In this case, when the end of the workfile (file #2) is reached, all the records transferred in the first part of the program have been printed out and we go to statements 550-570, which cause a line to be printed under the column of hours and then total hours under that. From there, calculations are done and the charges for time are printed after rounding the result to cents (580-640).

Finally, the workfile is again read; this time *just* for the expense items. These are printed and totaled; the total is added to the time-dollar total and the billing is closed.

Sample printouts of HRSCAL are shown in Fig. 6. Compare these to the input in Fig. 4 to see how the client-project separation and expense-item entry show up in the bill printouts.

General Micro-Translation Information

Herein find some general ideas about the differences between the BASIC used in these programs and the one you might have available.

I can't get multiple

statements per line on the IBM. You can, so use this feature—it saves memory and paper. Leave all the remarks out if you need even more room in memory. If you need more than 256 records per file, allocate all you need if your system permits (or less, if that is the case). If pressed for disk space, calculate the number of bytes per record you think you will actually take and use that figure in record-length allocation (if your system permits), instead of using the default size of 128 bytes.

Perhaps some of the formatting statements used will not be available to you. SKP (X) will cause the line feed to operate X number of times; SPA(X) causes the space function to operate X number of times; TAB(X) advances the carriage to the Xth space on the page. Knowing what these functions do will allow you to substitute what you have in place of them, if they are not available in your implementation of BASIC.

A colon (:) at the end of a PRINT line leaves my carriage at that point (i.e., no CR or LF is executed). Many microcomputer BASICs use a semicolon (;) for the same purpose.

The IMAGE statement causes numeric data to print with the numbers formatted in accordance with the pound signs. That is, if the statement is: 600 IMAGE "######.##"; a PRINT statement like : PRINT USING 600, D; will print the variable D with six numbers, a decimal point and two numbers. Leading zeros will be suppressed, and any numbers to the right of the hundredths column will be lost. For example, the number 1234567.904 would print as 234567.90. 456.5 would print as 456.50. If you don't have an IMAGE statement, again, you ought to be able to figure out how to use what you do have in-order to accomplish the same thing.

Most microcomputer BASICs are far more sophisticated than the one used on the IBM (i.e., such things as IF THEN ELSE are common). Make the most of what you have!

IF END . . . THEN . . .

Now, we get down to the real problem—the IF END statement. I certainly don't pretend to know much about all the microcomputer BASICs on the market. But, of those I am familiar with, none have the IF END statement. So, is all lost? No! I wouldn't have written this if that statement were essential. More important, the editors of *Kilobaud* wouldn't have printed it.

There are, in fact, several solutions that are even better than the IF END statement—solutions that will cut down search time by a factor of several hundred. There are other solutions that don't have the timesaving advantage, but will serve exactly the same purpose as the IF END statement.

First, let's look at substitutions for the IF END statement. You can write all zeros (or blanks, or whatever) into every record when you first set up the file. Of course, this will take a separate program, to be run when the file is first allocated. Be sure to use the same data type (string or numeric) for the variables you will eventually use, lest you get a weird result (or error message) when you run the program. If you are careful, you will have another program that checks to be sure that you have really written zeros into every record.

Then, you omit the IF END statement and, instead, ask the computer to continue its searching until it finds a record with the *first* data item a zero. This will be a blank record you can then write onto. Since you will overwrite the zeros with other data, that record will not be selected the next time you search.

The above, obviously, applies to HOURS. In HRSCAL, a similar scheme can be used to jump to the print portion of the program when you come to the first record with zeros in it—that will appear after all the records with *real* data have been read. The result of this is exactly the same as using the IF END statement, except that *you* have provided the end-of-file marker instead of having a built-in feature to do it for you.

You can also accomplish the same thing by writing, say, "9999" or "NNNN" into the record *after* the last one you use for valid data in HOURS. Then, naturally, you look for that variable content to find the place to write. This is an ancient trick, but is still valid, especially with tape.

A much more sophisticated and a great timesaving approach is to use indexing as an alternative to the IF END statement. In this method, you use the first record of a file to point to the last record used. That is, whenever you run HOURS, you first read record one (the index record), and then you set the record pointer to whatever number is contained in that record, plus one.

Just remember that *all* records in the file must contain the same data-types. The *first* item in your record must be a numeric variable (unlike the program listings) if you use an index key, as you will be comparing it to, and exchanging it with, another numeric variable (the record pointer). You will just have to lose the extra bytes that result, unless you also use client numbers. Also remember, unless you key to client numbers, to write a dummy numeric quantity into every record as the first item.

```
RECORDS READ  7
ANOTHER? Y
CL.NAME, PROJECT
? PETE JONES,TRUSTEE
RATE? 45


          ************************************************************
     ACCOUNT OF PETE JONES -- TRUSTEE AS OF 06/05/78

  DATE      SERVICE RENDERED                            BY       HOURS
1/12/77   TELCON JUDGE RE APPOINTMENT                   2         0.50
1/15/78   COURT HEARING                                 2         4.50
                                                                --------
                                          TOTAL HOURS             5.00

                                          TOTAL FEE IS      $   225.00

LIST OF EXPENSES:
   DATE     FOR                    BY           AMOUNT

                                          EXPENSES TOTAL $        0.00
                                                                --------
                                  TOTAL THIS STATEMENT      $   225.00

                         T H A N K   Y O U


          ************************************************************


RECORDS READ  7
ANOTHER? N

DONE
```

Fig. 6. Actual bills generated by HRSCAL. In actual use, the bill would be cut at the asterisks (*) and sent to client after it was copied. Compare with Fig. 4 to note the selection of items for printing (e.g., the first bill contains data from record #s 1, 4 and 7).

The most obvious advantage of the index method is that you will not have to read the first, say, 250 records to find that the first blank record is number 251. The example below shows some suggested code which would appear right before statement 110. R is the index.

```
80 LET C=1
85 READ #1,C;R
90 LET C=R+1
95 READ #1,C; R,A$,B$,M$,C$,P,D
```

(Use statements 110 and 120 as a check to be sure you really have a blank record.)

In the program HOURS, you would then go on to statement 130 and, therefore, write to the last record, plus one, that was in the file before entry. You get the idea by now, I'm sure: Use the first record to store a number that is the last record used for data. Then, use this to find a blank record on which to write new data. CAUTION: You have to write the number of the last record used *after* the execution of HOURS back into the index record before you exit the program.

And, remember, you make R equal to C *before* you reset C for a write to record one (the index record)—otherwise, you lose the count of the last record used!

```
295 Let R=C
300 Let C=1
305 Write 1, C; R
310 End
```

In HRSCAL, use the index record to trigger a jump when the record pointer reaches that number. Again, you will have to read in the contents of the index record (#1) before making the comparison.

Using the index method is really a great timesaver in a random-disk system. But, it has its own problems. Not all of these can be discussed in one article, but here is what I feel is the worst: If you want to delete a record from the file, you can, but that space will still not be available to you, as the index record will always point to the last record in the file that has been used. Deleted records will just be bypassed.

Solutions to this include periodically compressing the file to eliminate closed records and replacing them with active records; or a more complex index scheme that will point you to both blank and closed records, as well as the last available record in the file; or, if you will, adopting an I-don't-care attitude until the file is full and you *have* to care.

As was intimated before, the IF END use is an adoption of the I-don't-care attitude. When the file is full, you run bills, erase the file, and that is the end of that. A more complex attitude is just going to require more complex procedures—and some more companion programs.

Indexing also could be used to save the second pass through the workfile in HRSCAL—the pass used to find the expense items. Instead of just bypassing an expense record on the first pass, write its record number into an array. Then, when you are ready to print the expense items onto the bill-form, pull these numbers out of the array one at a time, set the record pointer equal to the number pulled out, and go right to the expense record—all without searching for an "E" in the date-field on the second pass.

Please note that any index approach *demands* random, disk hardware—tape is out, except for the most sophisticated systems. Note, too, that a complex index system (one using more than just a pointer to the last record used) is almost certainly going to require another file merely to hold the index records for pointing to closed, deleted, numbered or usable records in the master file. Confused? Well, you might be, at that.

My preliminary programs in Alpha-Micro BASIC for our complete, new system contain one file whose sole purpose is to index records contained in four other files, all of which contain information on the same client. It can get rather hairy! . . . and all because you don't have, or want to improve upon, the IF END statement. I will be happy to assist any of you trying to implement these programs on your computer. Please, though, give me enough information about your BASIC to do so, and include an SASE.

READ and WRITE

After reading your BASIC manual, you have probably come to the conclusion that the various READ and WRITE statements in these programs will not work in your system. Please, don't despair. Any disk or tape system worthy of the name will have *some* command to write information to a record and another to read data from that record. Just look under that section in your manual and modify the programs accordingly. The most common difference will be the lack of a record-pointer reference in the READ or WRITE statement itself. No problem. Leave it out and reference it earlier in the program.

It is unfortunate that no two implementations of BASIC are really identical. For example, the programs presented here will not run on any other large, mainframe computer, let alone on a microcomputer. But, if you know (or are informed about) what you need to look for, there should be little problem in adaptation.

Critical Comments

Most critical comments have already been made. I know that the programs as presented are far from the most sophisticated possible. I know that they both could be made more useful with a few rather minor changes.

For example, no provision is made for credits and payments. The programs assume one billing and one payment in full. For us, this is a valid assumption. If an hourly-billed client does not pay in full, he is in a past-due status and can easily be handled by hand billing. However, for many firms, this will not suffice. For these, the simple addition of several lines of code in both programs will take care of the problem. You could also use a separate file (BALFIL) (?) to keep track of the balances of each client.

Many readers will say that these programs are not *structured*. This is so true . . . sorry about that, really. Realize, please, that these programs are the efforts of a novice programmer. Many of you will have suggestions for modifications, improvements and so on. *Great!* . . . this is how the use of micros will grow. But, please, let *Kilobaud* know, OK? Improvements, modifications, doing the same thing in a better way—all this and more will cause hobbyist and business use of micros to grow. Again, please, let *Kilobaud* in on it. Wayne et al will smile, you might make a heap of money and, best of all for me, I'll profit from any critical comment.

Tape

I have worked with this medium only in a limited way—any comments herein should be taken with that in mind. In any event, I see no reason why these programs, with some modification, should not work on a tape-storage system. As presented, the programs are serial in nature, anyway. However, I can imagine that the search for records would be a terribly time-consuming task. If you can stand that, give it a try if you have at least two tape drives.

I suggest at least the following housekeeping programs and techniques.

1. The use of client numbers instead of names. This will simplify the tape system both as to comparison of records and ordering the records on a tape file.

2. Update and sort programs to keep search time down. You will need to have all records of a given client-project sequentially ordered on the tape.

3. A compression program to get rid of closed records. This is optional on disk, but almost mandatory on any tape system.

Disk: Support Programs

1. Backup—a simple copy program.

2. Delete—to get rid of records already billed. This can be either soft (where a unique key is used to replace the search-key) or hard (where the record is actually eliminated from the file).

3. Print-delete—a printout of all records deleted, before the actual deletion thereof; a copy of bills can serve this purpose, if HRSCAL is modified to delete the records it prints.

If you use client numbers, you should include in HOURS a check-digit routine to test for a valid number. Perhaps you would have use for another program (subroutine) to test for a duplicate number.

Security. If you modify the programs to show balances, then don't allow a client's record(s) to be closed out with an outstanding balance. Also, no credits (as contrasted to payments) should be allowed to any account without a security code. All this keeps operators honest.

Backup Files

I cannot stress too strongly the use of backup files. If you use these programs to any extent at all, you should make it a regular practice to copy the updated file onto another disk at regular intervals (i.e., daily, if there are many entries into the file each day). Then, if you bomb a file (and you will eventually), or if a disk goes bad for some reason, you will have the old one to turn to and only have to reenter a minimum of data to set things right. If you are going to offer a service to others, backup files are an absolute must!

Our firm pays lots of bucks because we insisted from the outset that a daily backup tape be made of our file—the extra CPU time is well spent. (Here, I'm talking about the main system: 1500 client records of 650 bytes each.) On five occasions to date, even with experienced operators, garbage has been written onto our disk file, totally destroying all the valuable data therein. It would have taken weeks to recreate our file by hand, if it could have been done at all. In each case, however, the computer center simply restored the data onto the disk by using the tape made the night before. We lost about 20 entries that had to be reentered.

On one memorable night, an operator with, maybe, more troubles on his mind than brains in his head, not only wrote garbage (the University Alumni list) into our current disk file, he also compounded the error by writing *that* onto our backup tape, thus destroying any hope of reconstruction—except for one thing: We also require the making of an end-of-month tape, saved until another one is made 30 days later. With this, we were able to restore the file, and we only had to reenter ten or twelve days' entries.

This was really a small price to pay compared to a complete recreation of the file and all the transactions that were recorded therein for the past two years. (The University spent about five weeks or so reconstructing the Alumni file, by hand, after their *only* tape of that file had *our* data written onto it in the same debacle—it was, to be sure, quite a night at the computer center.)

The moral of all this is obvious (especially if you are running a service for others): Make sure their files can be restored in the event of a goof or a system crash.

Summary

I hope this article has given you some food for thought, at least. These are working programs being used in my office to generate billing for which clients of mine respond with cash. None of these clients have given me any static about the novice nature of the programming—they have given me only money! Perhaps there is a lesson there.

I hope some of the ideas presented here, if not the specific programs, will assist you in your business or in selling a service to others who bill clients on an hourly basis. Let me, and *Kilobaud*, know how you come out.■

Troubleshooters' Guide

You're hesitant about tackling repair and interfacing problems? If so, this will point you in the proper direction and get you started.

Ralph Tenny
P O Box 545
Richardson TX 75080

One of the fascinating phenomena about computers is that they can do things—almost anything—*automatically*, provided the machinery to accomplish a given task is available and can be run and controlled by electrical signals. All such computer-controlled machines are called peripherals, no matter what their function is. Most of us routinely use such peripherals as TTYs or TVTs, audio cassettes and printers. Depending on whether you bought a system or built kits, you spent various amounts of time getting those peripherals to work with your computer.

You were probably furnished detailed instructions for operating the TTY or TVT and cassette with your computer, and the necessary electrical connections (interface circuitry) were already designed and ready to use. Finally, it is almost certain that the software for your computer already had provisions to operate the peripherals necessary to make the computer functional. These things are necessary for any computer system—interface, software and "how to"—but the information may not always be available when a peripheral made by one manufacturer is to be used with a computer from another manufacturer. In the case of surplus equipment such as a Baudot TTY, there may be no instructions or software available.

The Big Picture

Regardless of the circumstances, let's assume you are having trouble with a computer peripheral (otherwise why are you reading this?). The troubleshooting approach needed will vary with the type of computer and hardware that's involved.

The computers available to most hobbyists will be one of two types—those with isolated, or accumulator, input/output (I/O) such as the 8080 and 2650, or those with memory-mapped I/O such as the 6800 and 6502. Accumulator I/O machines have special input and output instructions, while the memory-mapped computers use standard memory instructions such as LOAD and STORE to service both memory and peripherals. In order for this to happen, such peripherals are assigned memory addresses; this will limit the total amount of memory available. However, since most microcomputers will address either 32K or 64K words of memory, it would take a lot of peripherals to make a dent in the available memory space!

The best troubleshooting method also depends upon which type of peripheral is involved—whether it is a device to input or output data or if it controls something. In general, controllers are a bit easier to troubleshoot because their input signals (combinations of bits on the input lines) are less numerous than those for data-handling devices. Also, it is likely that the interface connections will be simpler for the controller than for the data peripheral.

The final consideration will be software; detailed instructions must be available to enable the computer to produce the proper signals to drive any peripheral. Most manufacturers of hobbyist equipment furnish software for peripherals they produce. If surplus equipment is involved, there may be no software available unless someone has also made a kit available to interface the machine to a microcomputer. Note that if appropriate software is available, it will depend strictly upon a properly functioning interface of a particular design.

Successful troubleshooting of computer peripherals requires careful study of the system—from microcomputer architecture through the mechanical and electrical details of the peripheral and interface. You are probably familiar with your computer, so let's begin with the peripheral, giving it a thorough inspection to be sure it is totally functional. Operate all controls, and supply electrical signals if appropriate. Proceed to the interface circuitry only if these tests are successful.

Be sure the interface is capable of reliably and safely producing all the signals needed by the peripheral. Relay contacts or semiconductor switches (power transistors, SCRs or Triacs) must be able to handle the voltages and currents involved. A stuck relay or defec-

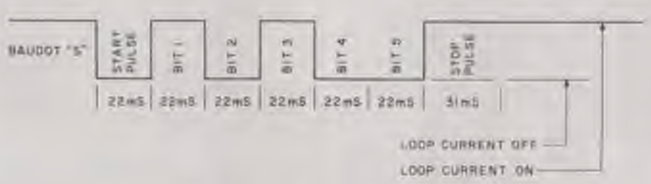


Fig. 1. Baudot code timing.

HEX CHARACTER	BAUDOT	HEX CHARACTER	BAUDOT
0	01101	8	01100
1	11101	9	00011
2	11001	A	11000
3	10000	B	10011
4	01010	C	01110
5	00001	D	10010
6	10101	E	10000
7	11100	F	10110
CARRIAGE RETURN	00010	LINE FEED	01000
SPACE	00100	LETTERS	11111
FIGURES	11011		

Fig. 2. Chart of some Baudot codes.

tive Triac won't hack it—all systems must be GO! Power supplies must be able to handle the load. If the computer power supply is used, be sure there is reserve capacity and that the computer power bus isn't receiving power glitches caused by the peripheral.

Now, consider the software. Let's take the case where the software is for a different computer system. Break it down into modules so that only one peripheral operation at a time is addressed. Determine what changes must be made to perform the same function on your computer, and make changes as necessary. Finally, combine the modified modules to check out the entire system—computer port, interface and peripheral—in a simple loop or repetitive fashion that addresses all desired peripheral functions. At this point, you should have mastered the system well enough so you can create application software to make the peripheral do useful tasks as a part of the whole system.

Serial Data Testing

Now, let's examine an actual problem. My KIM-1 has software (monitor ROM) and an interface to drive an ASR-33 Teletype directly; but how about operating my Model 15 Baudot TTY? All I have is documentation for KIM and a service manual for the Model 15—no software and no interface circuitry. I want the Model 15 to serve only as a printer, so there is no need to interface the keyboard; KIM would drive the printer mechanism using whatever data I wish to feed it. The keyboard was useful in checking out the machine initially, but the interface will be much simpler if the keyboard is not involved.

The first step in planning this project is to understand how the machine works. The print mechanism operates when current through a selector magnet is interrupted in a certain code pattern. Fig. 1 is a timing diagram of the Baudot code format, and Fig. 2 is a chart of Baudot codes for the hexadecimal and TTY control characters needed. Note that a Baudot machine has no SHIFT key, but that it has LETters and FIGures keys. (The five-level code will select only 32 keys, but by arranging for numbers, symbols and punctuation marks to be uppercase—FIGures mode— the print set is expanded to 58 codes.)

Look again at Fig.1 and note that the Baudot code consists of a start bit, five code bits and an extra-length stop bit. That stop bit could be troublesome to make, so let's modify the code format as shown in Fig. 3—that is, substitute two regular-length stop bits for the longer one. Previously (Fig. 1), the character time was 163 ms; now it is 176 ms—only 8 percent slower. The resulting simplification of software and hardware makes the trade-off entirely acceptable.

Now, how can KIM drive the Model 15? KIM has a 20 mA current loop derived from a 5 V supply, which is entirely inadequate to switch the 60 mA current derived from the high-voltage supply of the Model 15 (Fig. 4). Of course, it is possible to add some external circuitry to adapt KIM to Model 15, but KIM's software is still a problem. KIM's lookup table is for hexadecimal to ASCII, and the self-adjusting timer, which produces the proper output bit rate, requires the keyboard for setup. It is easier to build a simple interface for the Model 15 and drive it from the standard KIM output lines.

Fig. 5 is a simple CMOS circuit that accepts parallel data and will shift it out to the Model 15 through two transistors. Q1 matches the CMOS output to the higher drive requirements of Q2, which is a high-voltage unit to switch the current loop. IC1 is a shift register, and IC2A is a free-running oscillator with a period of 22 ms. IC2B is a pulse generator that causes the shift register to load data from the five input switches.

A cycle of operation works this way: push-button switch SwA trips IC2B, and data is loaded into IC1, including a 1 on pins 1 and 7 and 0 on pin 15. As soon as the load pulse terminates, data is ready to shift out as clocked by IC1A. When the 0 loaded by pin 15 reaches the output, Q1 and Q2 turn off, producing a start pulse for the Model 15. As the remainder of the data is shifted out, 1s are shifted in via pin 11, the serial input pin. Thus, after loaded data is shifted out, Q1 and Q2 remain turned on, waiting for the next data to be loaded.

The proper test method for this interface is to set the switches to the Baudot codes for R (01010) and Y (10101) alternately. Carefully adjust the frequency of IC2A until proper operation is obtained, then try other characters. The circuit in

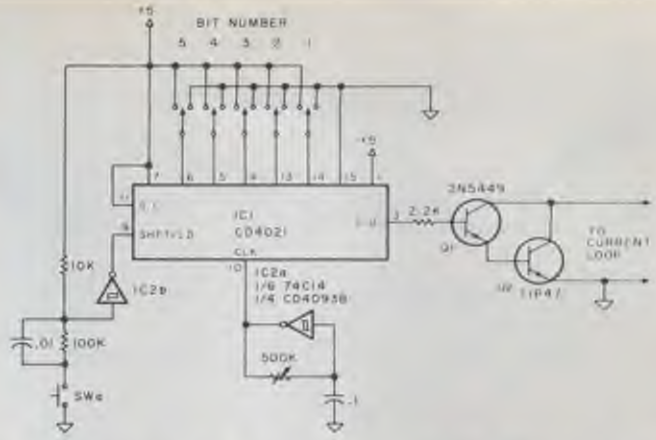


Fig. 5. Parallel-to-serial driver for Model 15 TTY (with test switches).

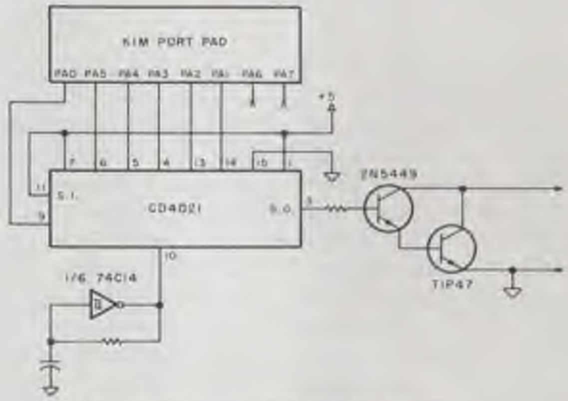


Fig. 6. Completed Model 15 interface.

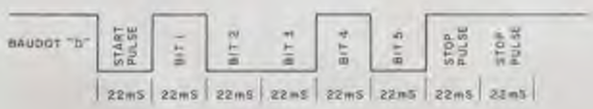


Fig. 3. Modified Baudot code timing.

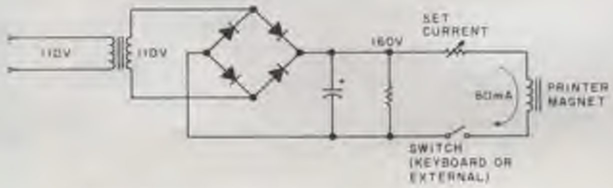


Fig. 4. Common Model 15 current-loop supply.

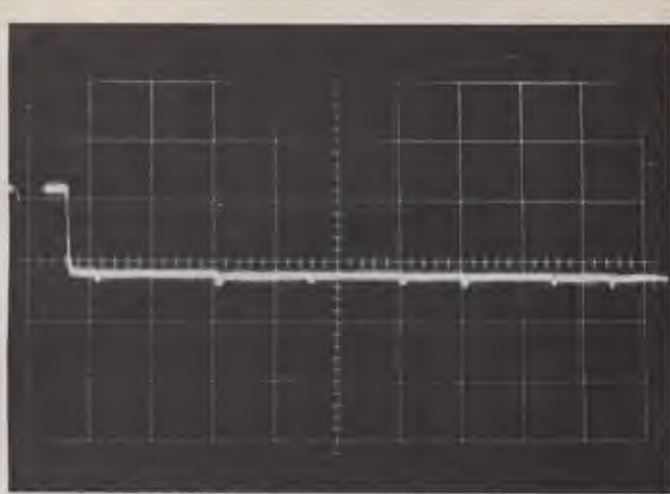


Photo 1. Instruction (LDA 8000) causes Address Bit A15 to go high as the instruction is executed; this defines a unique strobe to furnish sync for an oscilloscope.

```
Label   Op.  Arg.       Comments
TEST    LDA  #$3F       Select mask for port.
        STA  PADD       Set mask in port control register.
        LDX  #$00       Zero index register as counter.
PRINT   LDA  BUFFER, X  Get first data byte.
        STA  PAD        Set data in port.
STROBE  INC  PAD        Toggle PA0 to load data
        DEC  PAD        into interface.
        INX             Set index for next data byte.
        CPX  END        Test for last byte.
        BEQ  OUT        Done? If so, exit.
WAIT    LDA  #$F0       Set time in
        STA  TIMER      programmable timer.
TIME    BIT  TIMER      Timer done?
        BPL  TIME       No, go back and check again.
        BMI  PRINT      Yes, print again.
OUT     BRK             Stop computer after printing last byte.
```

Fig. 8. Short test program writes data from location BUFFER to interface.

Fig. 5 has now become a tested interface for the Model 15.

Although it was designed with KIM in mind, it can easily be adapted to any computer. Fig. 6 shows the KIM output port attached to the interface; note that PA0 (least significant bit of the port) replaces IC2B. Data can be written out to the port on pins PA1 through PA5, and the PA0 can be toggled (turned on and off) to load the data.

Software to operate the interface can take many forms, depending upon how the printer needs to work with a main program. For checkout, a short test routine is best. Fig. 7 is a flowchart and Fig. 8 is the KIM program. Note how the software is intimately related to the hardware.

Begin with the pin assignments of the output port: PA0 is the least significant bit (LSB), so it can be toggled with INCrement and DECrement instructions. The data buffer has the organization shown in Fig. 9, where X stands for "don't care," or unused, bits. Bit 0 (LSB) is always 0, so the load/shift pin of IC1 is always low for shifting until new data is to be loaded.

Let's follow through the program after a brief comment on parts of the setup. At label TEST, 3F is loaded to the port. Ones loaded into the Data Direction register (PADD) turn the corresponding port lines into outputs, while 0s create inputs. So 3F makes bits PA0 through PA5 outputs as required for the interface. Output data is stored at a group of addresses named BUFFER, and the location named END contains data specifying how many words of data BUFFER contains.

The location named TIMER is a programmable timer that sets bit PB7 low when time is up. So, when the progam is entered at TEST, the port is set up and register X is zeroed to make a counter. Data is loaded at PRINT, stored, and then IC1 is loaded by STROBE. Register X is incremented and tested; the routine at WAIT marks time until IC1 completes a print cycle. If the last byte has been printed, the BRK instruction at OUT stops the computer. By now, it is even more apparent that software and hardware must work in exact harmony if any computer-controlled task is to be successfully completed!

The example above may seem contrived and simple, but it illustrates the most important points about debugging peripherals and their interface circuitry: *Never* try to debug malfunctioning equipment with an applications program. Always break up the task into as many modules as possible. It's OK to fire up a peripheral with furnished software if and *only* if you are dealing with a turnkey package in which the software and hardware were created for each other. *Then* if it doesn't run, follow the suggestions and examples above.

Let's return to the point where manual and electrical testing of a peripheral seems to prove the mechanism functional. I will assume that you understand the signals and power that must be furnished by the interface. If you don't, stop until you find out! If the interface furnishes power, substitute a similar load and write a

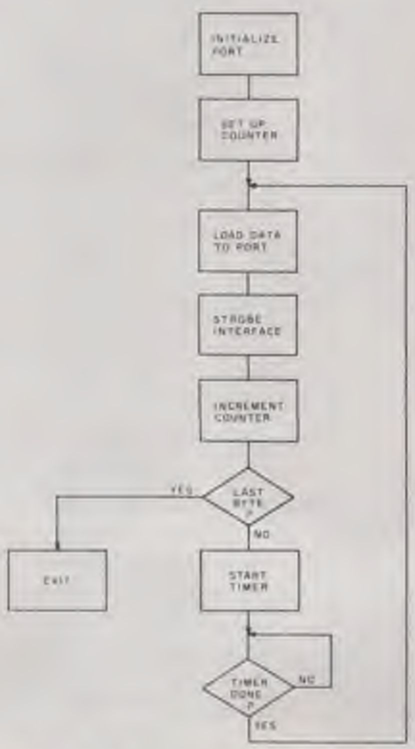


Fig. 7. Flowchart for test program.

MSB							LSB
X	X	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	0

Fig. 9. Data buffer organization for loading Model 15 interface.

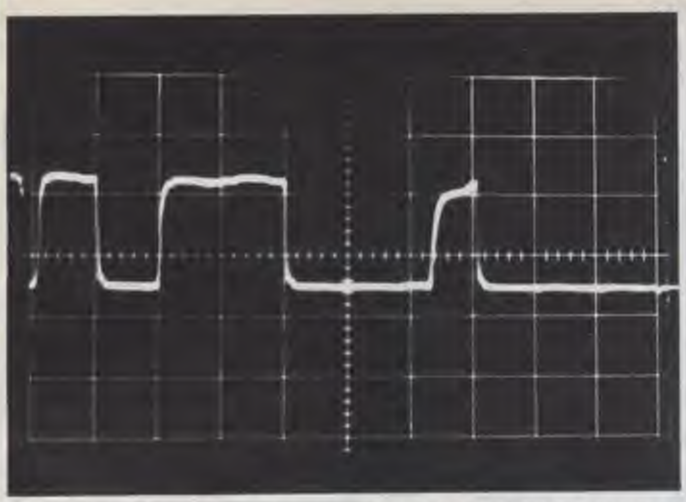


Photo 2. Using the sync shown in Photo 1, Data Bit 2 is examined for proper data activity and timing (see text for details).

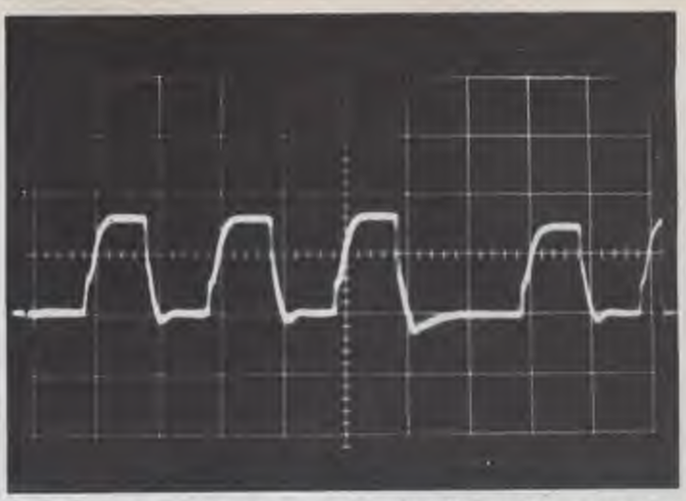


Photo 3. Heavy capacitative loading of buffered address line produces slow-rising leading edges and undershoot on trailing edges.

short program that turns on the power and then halts the computer (or loops while waiting for a Reset signal). Measure the voltage across the substituted load to be certain power is really being delivered. Check for switch closures with an ohmmeter. Do each function of the interface the same way until it all checks out.

Scope Techniques

Data transfers or special signal patterns are harder to check. The *neat* way to do this is to use a digital analyzer—about $2K worth. Not too many of us have one, so there must be another way. The next easiest way to troubleshoot is to use an oscilloscope. Due to the short time any particular word of data stays on the data bus—just one part of a machine cycle, which may be less than a microsecond—the scope must have a triggered sweep. Even then, it is difficult to find a trigger signal that triggers only when (or just before) the data to be checked appears on the data bus. It is sometimes possible to make the computer generate a unique strobe shortly before outputting a data byte. For example, Fig. 10 is a short program that generates a unique address (LDA 8000 sets bit A15 high) and then moves data to the address bus, generates a short delay and repeats. When the scope is triggered by bit A15 going high, it is possible to examine the data bus bit by bit and verify that the proper data is appearing.

Photo 1 shows bit A15 of the address bus going high at the first statement of the program in Fig. 10 (LDA #8000). When this signal is used as a trigger for the scope, it is possible to examine the data bus and watch for the data to appear. This is shown in Photo 2. The exact details of what follows will be pertinent only to the MCS 6502, which is the processor used in the KIM-1 microcomputer, and a similar analysis will have to be made for each different uP.

Label	Op.	Arg.	Comments
TRIG	LDA	$8000	Set address bit A15 high as strobe.
	LDA	#$A5	Get data for output.
	STA	PORT	Send data to port.
	LDX	#$F0	Set up index register as counter.
COUNT	DEX		Decrement counter.
	BNE	COUNT	Loop back if counter not zero.
	BEQ	TRIG	Start over after counter reaches zero.

Fig. 10. Short program generates scope trigger to verify data movement.

In Photos 1 and 2, the timebase speed is such that one machine cycle takes one horizontal division on the scope face. Keep that in mind, and it will then be possible to analyze when the data should appear on the data bus. The next step is to count machine cycles through the program (see Example 1).

Since the absolute address appears only in the last cycle of the first instruction, Photo 1 shows that ending cycle. Counting forward six more cycles (six divisions on the scope graticule in Photo 2), we see that a data bit comes high in that cycle. The subject of Photo 2 is Data Bit line 2, which should be a 1 according to the data loaded in instruction 2. The bit pattern with a data byte of A5 is shown in Example 2. Thus we see that Data Bit 2 *should* be a 1; checking other bits on the data bus showed that the correct data was appearing at each pin.

To summarize the procedure for checking data (or address) bit on computer bus lines, begin by creating a software strobe or other means of sync for the scope. Set the scope time base so that one machine cycle occupies one horizontal division of the scope graticule. Analyze the program to determine the number of machine cycles required to bring the data to the bus; remember that this analysis depends upon

Instruction	Type Instruction	Cycles
1. LDA $8000	Load accumulator absolute	4 (use last cycle only)
2. LDA #$A5	Load accumulator immediate	2
3. STA Port	Store accumulator absolute	4
		7

Example 1.

Data Bit	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Binary value	1	0	1	0	0	1	0	1

Example 2.

Address lines	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Address 10AF	X	X	O	N	X	O	O	X	N	O	N	O	N	N	N	N
0F36	X	X	O	O	N	N	N	N	O	O	N	N	O	N	N	O
3668	X	X	N	N	O	N	N	O	O	N	N	O	N	O	O	X

Example 3.

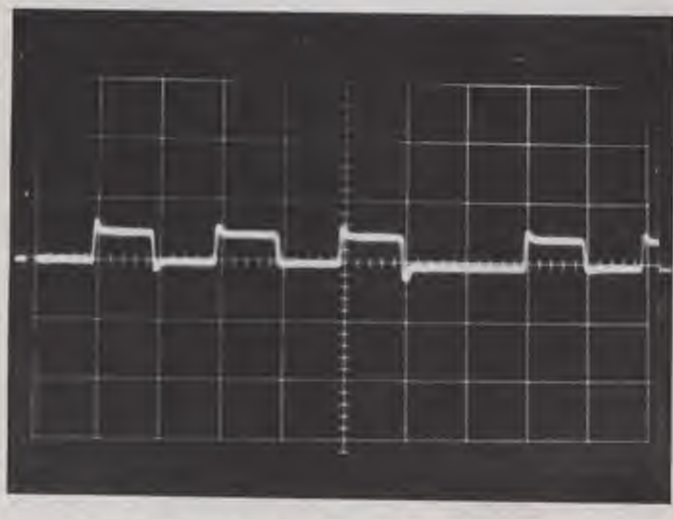


Photo 4. High resistive loading (partial short to ground) reduces address line amplitude to below logic 0 levels.

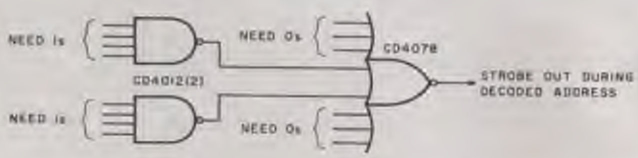


Fig. 11. Selective decoding gives unique event to check program branching.

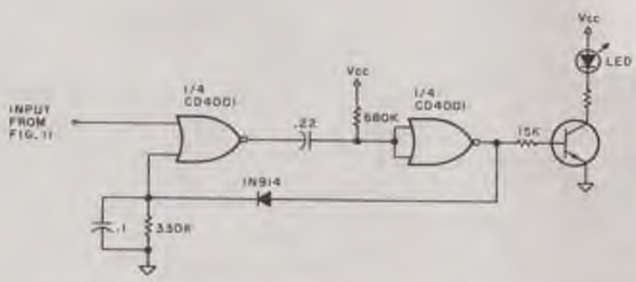


Fig. 12. Simple CMOS one-shot makes LED blink slowly from repetitive triggers.

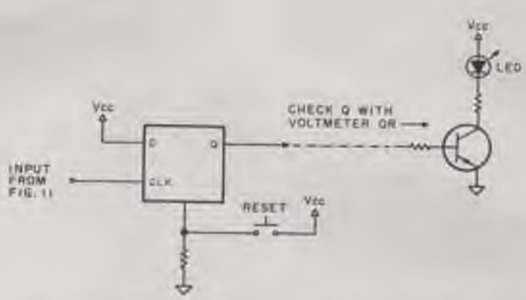


Fig. 13. Flip-flop records infrequent triggers.

thorough knowledge of the machine timing and the number of cycles each instruction takes. Finally, determine where (on the scope graticule) the data should appear, and look at each line on the bus to verify correct operation.

Photo 3 shows a buffered address line that has heavy capacitive loading. Note that rising edges are heavily rounded, and that falling edges have undershoot with a slow recovery. Photo 4 shows the same address line with a heavy resistive load; note that the amplitude is drastically reduced and obviously will not meet the voltage levels required to operate either TTL or MOS circuits. Any device driven by the line shown in Photo 3 will probably show erratic or false address decoding; the waveform of Photo 4 will probably cause a driven device to consider this address bit a 0 at all times. In either case, faulty addressing will be the symptom and a scope would be needed for proper diagnosis.

Other Tools

It is often possible to use ingenuity and planning to do much troubleshooting with a voltmeter or other static indicator. Also, if your system has a front panel with address switches and status lights, the pattern on the lights may offer helpful hints. If data fails to appear at the expected place, it can be very helpful to know that the subroutine that moves the data wasn't called by the main program. How? If the subroutine happens to be in a little-used page of memory, sometimes it is possible to see the address LEDs on the front panel flicker as the subroutine is accessed. A program loop can be used to enhance the brightness of the LEDs.

In a similar vein, checking certain address bits with a logic probe (a "pulse catching" feature is necessary) can reveal that the computer is accessing certain parts of memory. If no single unique address bit is involved, a simple two-IC circuit (Fig. 11) will decode enough of the address to generate a unique pulse each time the memory accesses the decoded address. This pulse will then trigger the logic probe so that it blinks. If a logic probe is not available, hook up a one-shot (Fig. 12), which will make a blinking light, and trigger it from the decoder of Fig. 11.

The decoder of Fig. 11 works this way: It can decode 14 address lines, but the choice of these lines will depend both on the address to be decoded and on other parts of the program with similar addresses to be excluded. Example 3 shows some hex addresses and possible decoding connection choices. N represents a NAND input line, O is a NOR connection, and X indicates lines left open.

The object is to make the best use of the available gate inputs so that only addresses within the subroutine are decoded, while no addresses in the main program are decoded. This will ensure that the decoder will develop an output only when the subroutine is addressed.

Finally, a simple flip-flop (Fig. 13) can be triggered by the address decoder. This is particularly useful for checking on events that happen infrequently, such as monitoring switch closures. Select an address within the program section that reacts to the switch closure; then close and open the switch. If the flip-flop is set by the decoder, all is well. Use a voltmeter to check the Q output of the flip-flop, or use a transistor driver to turn on an LED. Check both states—be sure the flip-flop is set by the computer's response to the switch closure and that it stays reset as long as the switch stays open.■

Metric-American Conversion Program

With September comes the opening of schools. Here's a program to help students with the inevitable: learning the metric system.

Mickey Ferguson
P. O. Box 708
Trenton GA 30752

I'll never understand women! If I live to be 100, I'll just never understand women. Women in general are difficult, but the lady to whom I'm married is impossible! Let me tell you a little about her. Foxy is a caver. Now, cavers are really strange people — I mean *really strange*. Stranger even than ham radio operators or computer hobbyists! (Rumor has it that, as a body, their most notable accomplishment is the origination of the EAT MORE POSSUM bumper sticker.) I would even go so far as to say they're addicts! Get them too far into the daylight and away from bat guano and they start having withdrawal pains! When they venture out from underground, they do many weird things to support their caving habits.

Some cavers are college professors, others run companies such as Western Electric, while others are directly responsible for putting men on the moon. I've also heard of one who writes "cookbooks" (like TTL, CMOS, & TV Typewriter Cookbooks). Well, Foxy was a professional computer person for a number of years and has taught me the little I know about programming. But, above all, Foxy is a caver. If you happen to see a short, blonde lady wearing mountaineering boots, grubby jeans, a T-shirt with pictures of bats all over the back, and who has a furry little rug-rat (Yorkshire Terrier) or two in tow; it's probably her.

Well, I've learned to live in a world of Bluewater (a rope), Break-bar racks (thingamajigs for going down rope), jumars (whatsits for going back up rope), carbide lights, occasional meals that smell like carbide, etc. I've even learned to live with Land Rovers full of total strangers appearing at our front door at three in the morning looking for a place to "crash" for the night. Trenton, Georgia, may not be the crossroads of the world, but it is the heart of cave country in the Southeast. In fact, that's why we live here, and this does tend to make life interesting. You simply never know just who will come visiting. Recently, I had the pleasure of demonstrating my M6800 system to a computer PhD from one of our large universities who had never even heard of microprocessors! I still chuckle when I think of him standing there in unabashed, openmouthed amazement at the small size and vast power of the system.

The other day, I walked into our study, where Foxy was busily preoccupied with her Great Unfinished Masterpiece. You see, she is writing a book for the National Speleological Society (the national U.S. caving club). The book, as Johnny Carson might say, contains *everything* that anyone could ever hope to know about descending a rope. Just what good it does you to know that is totally beyond me. So, there she sat, surrounded by mountains of paper, a pencil between her teeth, one of our rug-rats on her lap, punching away angrily on a pocket calculator, and muttering softly to herself.

"Hi, darling!" said I.

"Wha? Oh, G'way. Um bithy," she replied.

"Take the pencil out of your mouth, dear," I requested.

"Oh, hon, stick BASIC in the machine and write me a little program to convert anything to anything," she said.

"Huh!?!" I responded, preparing to make a quick exit.

To make a long story short, after several more minutes of this intellectually stimulating discourse, I discovered that she had been trying to do a lot of conversions of American weights and measures to the metric variety, making a lot of mis-

takes in the process. After all, her book is a heavy scientific (?) work that she feels deserves all references to weights and measures in both systems. I'll never understand women! There she sits, this lady who has forgotten more about computer programming than I'll probably ever know, pounding away on an old pocket calculator attempting to process a large volume of data with a perfectly good computer across the room doing absolutely nothing! Now she wants me to write a program for her that she could do better and faster herself. As I stood pondering all of this, she kept prodding me to do it. I could only think of one excuse: I told her that I was late for an appointment at the local massage parlor. She quickly pointed out, however, that we don't have a local massage parlor. My mind went blank; I had lost and knew it.

The Objectives

I set about the task of writing the Metric-American Conversion program with a few thoughts firmly in mind. First, I wanted to write it with a structured approach, as I've been reading so much about in *Kilobaud*. By doing it this way, it should be easy for anyone to understand (and, if desired, modify) just by reading the listing. Additionally, it should run in almost any BASIC with little or no modification. This should make it easy to write an article about the program, which (maybe, just maybe) Wayne would be willing enough to buy for *Kilobaud* or *73*. (I usually write programs with this in mind because I'm trying to save enough for a floppy disk system.) The other thing I had in mind was that it should be as idiot-proof as possible — because Foxy would be using it! (And, if I don't explain the preceding sentence, I'll be sleeping in the guest room for the next several months!) You see, when Foxy does anything

```
0001 REM *****************************************
0002 REM * METRIC-AMERICAN CONVERSION PROGRAM *
0003 REM *                BY                     *
0004 REM *          MICKEY E. FERGUSON           *
0005 REM *****************************************
0006 REM
0010 GOSUB 230
0020 PRINT TAB (10); "METRIC-AMERICAN CONVERSION"
0030 PRINT
0040 GOSUB 270
0050 PRINT TAB(5);"1. . . LENGTH"
0060 PRINT TAB(5);"2. . . AREA"
0070 PRINT TAB(5);"3. . . VOLUME"
0080 PRINT TAB(5);"4. . . WEIGHT"
0090 PRINT TAB(5);"5. . . TEMPERATURE"
0100 PRINT
0110 PRINT "PLEASE INPUT THE 'NUMBER' OF YOUR CHOICE."
0120 PRINT TAB(5);"(INPUT '99' TO END)"
0130 INPUT X
0140 IF X<1 THEN 10
0150 IF X=99 THEN 9999
0160 IF X>5 THEN 10
0170 ON INT(X) GOSUB 290, 730, 1170, 1610, 2050
0180 PRINT
0190 PRINT "HAVE YOU FINISHED     (YES=1)";
0200 INPUT X
0210 IF X=1 THEN 9999
0220 GOTO 10
0225 REM ***THE FOLLOWING SUBROUTINE ERASES THE SCREEN ON MY TERMINAL***
0226 REM ***IT DOES THIS BY PRINTING LINEFEEDS & CARRIAGE RETURNS***
0230 FOR X=1 TO 20
0240 PRINT
0250 NEXT X
0260 RETURN
0265 REM ***THE FOLLOWING SUBROUTINE PRINTS A STRING THAT IS USED IN***
0266 REM ***MANY DIFFERENT PLACES IN THE PROGRAM***
0270 PRINT "DO YOU WANT TO CONVERT:"
0280 RETURN
0285 REM ***LINE 290 IS THE ENTRY OF LENGTH CONVERSION SUBROUTINE***
0290 GOSUB 230
0300 PRINT TAB(10);"LENGTH CONVERSION"
0310 PRINT
0320 GOSUB 270
0330 PRINT TAB(5);"1. . . KILOMETERS TO MILES"
0340 PRINT TAB(5);"2. . . METERS TO FEET"
0350 PRINT TAB(5);"3. . . CENTIMETERS TO INCHES"
0360 PRINT TAB(5);"4. . . MILES TO KILOMETERS"
0370 PRINT TAB(5);"5. . . FEET TO METERS"
0380 PRINT TAB(5);"6. . . INCHES TO CENTIMETERS"
0390 INPUT X
0400 IF X<1 THEN 290
0410 IF X>6 THEN 290
0420 ON INT(X) GOTO 430,480,530,580,630,680
0430 PRINT "KILOMETERS";
0440 INPUT K
0450 M=K/1.6094319
0460 PRINT K;"KILOMETERS = ;M;"MILES."
0470 GOTO 720
0480 PRINT "METERS";
0490 INPUT M
0500 F=M/.3048006
0510 PRINT M; "METERS = ";F;"FEET."
0520 GOTO 720
0530 PRINT "CENTIMETERS";
0540 INPUT C
0550 I=C/2.540005
0560 PRINT C;"CENTIMETERS = ";I;"INCHES."
0570 GOTO 720
0580 PRINT "MILES";
0590 INPUT M
0600 K=M*1.6094319
0610 PRINT M;"MILES = ";K;"KILOMETERS."
0620 GOTO 720
0630 PRINT "FEET";
0640 INPUT F
0650 M=F*.3048006
0660 PRINT F;"FEET = ";M"METERS."
0670 GOTO 720
680 PRINT "INCHES";
0690 INPUT I
0700 C=I*2.540005
0710 PRINT I;"INCHES = ";C"CENTIMETERS."
0720 RETURN
0725 REM ***LINE 730 IS THE ENTRY OF THE AREA CONVERSION SUBROUTINE***
0730 GOSUB 230
0740 PRINT TAB(10);"AREA CONVERSION"
0750 PRINT
0760 GOSUB 270
0770 PRINT TAB (5);"1. . . SQ KILOMETERS TO SQ MILES"
0780 PRINT TAB(5);"2. . . SQ METERS TO SQ FEET"
```

```
0790 PRINT TAB(5);"3. . . SQ CENTIMETERS TO SQ INCHES"
0800 PRINT TAB(5);"4. . . SQ MILES TO SQ KILOMETERS"
0810 PRINT TAB(5);"5. . . SQ FEET TO SQ METERS"
0820 PRINT TAB(5);"6. . . SQ INCHES TO SQ CENTIMETERS"
0830 INPUT X
0840 IF X<1 THEN 730
0850 IF X>6 THEN 730
0860 ON INT(X) GOTO 870,920,970,1020,1070,1120
0870 PRINT "SQ KILOMETERS";
0880 INPUT K
0890 M = K/2.5899985
0900 PRINT K;"SQ KILOMETERS = ";M;"SQ MILES"
0910 GOTO 1160
0920 PRINT "SQ METERS";
0930 INPUT M
0940 F = M/.0929034
0950   PRINT M;"SQ METERS = ";F;"SQ FEET"
0960 GOTO 1160
0970 PRINT "SQ CENTIMETERS";
0980 INPUT C
0990 I = C/6.451626
1000 PRINT C,"SQ CENTIMETERS = ";I;"SQ INCHES"
1010 GOTO 1160
1020 PRINT "SQ MILES";
1030 INPUT M
1040 K = M*2.5899985
1050 PRINT M;"SQ MILES = ";K;"SQ KILOMETERS"
1060 GOTO 1160
1070 PRINT "SQ FEET";
1080 INPUT F
1090 M = F*.0929034
1100 PRINT F;"SQ FEET = ";M;"SQ METERS"
1110 GOTO 1160
1120 PRINT "SQ INCHES";
1130 INPUT I
1140 C = I*6.451626
1150 PRINT I;"SQ INCHES = ";C;"SQ CENTIMETERS"
1160 RETURN
1165 REM ***LINE 1170 IS THE ENTRY OF THE VOLUME CONVERSION SUBROUTINE***
1170 GOSUB 230
1180 PRINT TAB(10);"VOLUME CONVERSION"
1190 PRINT
1200 GOSUB 270
1210 PRINT TAB(5);"1. . . CUBIC CENTIMETERS TO CUBIC INCHES"
1220 PRINT TAB(5);"2. . . CUBIC METERS TO CUBIC FEET"
1230 PRINT TAB(5);"3. . . LITERS TO GALLONS"
1240 PRINT TAB(5);"4. . . CUBIC INCHES TO CUBIC CENTIMETERS"
1250 PRINT TAB(5);"5. . . CUBIC FEET TO CUBIC METERS"
1260 PRINT TAB(5);"6. . . GALLONS TO LITERS"
1270 INPUT X
1280 IF X<1 THEN 1170
1290 IF X>6 THEN 1170
1300 ON INT(X)) GOTO 1310,1360,1410,1460,1510,1560
1310 PRINT "CUBIC CENTIMETERS";
1320 INPUT C
1330 I = C/16.387156
1340 PRINT C;"CUBIC CENTIMETERS = ";I;"CUBIC INCHES"
1350 GOTO 1600
1360 PRINT "CUBIC METERS";
1370 INPUT M
1380 F = M/.028317
1390 PRINT M;"CUBIC METERS = ";F;"CUBIC FEET"
1400 GOTO 1600
1410 PRINT "LITERS";
1420 INPUT L
1430 G = L/3.785332
1440 PRINT L;"LITERS = ";G;"GALLONS"
1450 GOTO 1600
1460 PRINT "CUBIC INCHES";
1470 INPUT I
1480 C = I*16.387156
1490 PRINT I;"CUBIC INCHES = ";C;"CUBIC CENTIMETERS"
1500 GOTO 1600
1510 PRINT "CUBIC FEET";
1520 INPUT F
1530 M = F*.028317
1540 PRINT F;"CUBIC FEET = ";M;"CUBIC METERS"
1550 GOTO 1600
1560 PRINT "GALLONS";
1570 INPUT G
1580 L = G*3.785332
1590 PRINT G;"GALLONS = ";L;"LITERS"
1600 RETURN
1605 REM ***LINE 1610 IS THE ENTRY OF THE WEIGHT CONVERSION SUBROUTINE***
1610 GOSUB 230
1620 PRINT TAB(10);"WEIGHT CONVERSION"
1630 PRINT
1640 GOSUB 270
1650 PRINT TAB(5);"1. . . GRAMS TO OUNCES"
1660 PRINT TAB(5);"2. . . KILOGRAMS TO POUNDS"
1670 PRINT TAB(5);"3. . . METRIC TONS TO TONS"
```

sitting down, she has at least one rug-rat in her lap. Rug-rats are funny little animals. They don't ask for attention; they demand it! They will do strange things when you least expect it, like standing on your terminal's keyboard. So, our programs have to be rug-rat proof. Whew!

I think I've succeeded rather well in meeting my objectives with this program (especially if *Kilobaud* does buy this effort of mine). Although the Metric-American Conversion program looks rather long, it will run in a 12K system when used with SWTPC 8K BASIC – or even less if you leave out the REM statements. (REM does mean remove, doesn't it?) SWTPC 4K BASIC will also run it with no modification, and so should whatever BASIC you're using (assuming it has floating point arithmetic). The only things in the program that might cause you a spot of bother are the ON . . . GOTO and ON . . . GOSUB statements. In some implementations of BASIC, these will have to be changed to GOTO . . . OF and GOSUB . . . OF in order for the program to run properly.

The Program

The Metic-American Conversion program can be broken down into several subprograms (which are written as subroutines) that actually handle all of the conversion. These subprograms are tied together by a very simple control program that is written as a loop. I'm not too good at hieroglyphics (flowcharts) so I'll try to give you a basic outline of how the program works.

First, the control loop:

1. Clear screen on CRT.
2. Print a list of available conversion routines for the human.
3. Get an input from the human.
4. If input is invalid, go to #1 above.
5. If input is valid, go to

conversion subroutine.

6. Ask the human if done.

7. Get input from the human.

8. If human is done, end.

9. If human is not done, go to #1 above.

The conversion subroutines are, in outline, all identical. They are:

A. Clear screen on CRT.

B. Print list of conversions available in subroutine for human.

C. Get input from the human.

D. If input invalid, go to # A, above.

E. If input is valid:

1. Ask the human for the data to be converted.
2. Get input from the human.
3. Do conversion.
4. Print answer for the human.
5. Return from the subroutine.

Referring to the sample run of the Metric-American Conversion program, you see the program will convert measurements of length, area, volume, weight and temperature from metric to American or from American to metric systems of measurement. In the sample run, we chose to do a length conversion of one inch to centimeters. As you can see, the length conversion does not allow yards or millimeters, which are both common measurements. This is because I have no difficulty in converting feet to yards or yards to feet, nor do I have any problem with centimeters to millimeters or millimeters to centimeters. And I seriously doubt if you do either. Also, we have so very many different kinds of volume measurement (cubic inches, cubic feet, cubic yards, fluid ounces, pints, quarts, gallons, bushels, pecks, etc., etc.) that I was forced to choose only those I considered most important. So, you may wish to add or substitute those you consider most useful of the volume measurements.

```
1680 PRINT TAB(5);"4. . . OUNCES TO GRAMS"
1690 PRINT TAB(5);"5. . . POUNDS TO KILOGRAMS"
1700 PRINT TAB(5);"6. . . TONS TO METRIC TONS"
1710 INPUT X
1720 IF X <1 THEN 1610
1730 IF X >6 THEN 1610
1740 ON INT(X) GOTO 1750,1800,1850,1900,1950,2000
1750 PRINT "GRAMS";
1760 INPUT G
1770 O = G/28.349527
1780 PRINT G;"GRAMS = ";O;"OUNCES"
1790 GOTO 2040
1800 PRINT "KILOGRAMS";
1810 INPUT K
1820 P = K/.4535924
1830 PRINT K;"KILOGRAMS = ";P."POUNDS"
1840 GOTO 2040
1850 PRINT "METRIC TONS";
1860 INPUT M
1870 T = M/.9071849
1880 PRINT M;"METRIC TONS = ";T;"TONS"
1890 GOTO 2040
1900 PRINT "OUNCES";
1910 INPUT O
1920 G = O*28.349527
1930 PRINT O;"OUNCES = ";G;"GRAMS"
1940 GOTO 2040
1950 PRINT "POUNDS";
1960 INPUT P
1970 K = P*.4535924
1980 PRINT P;"POUNDS = ";K;"KILOGRAMS"
1990 GOTO 2040
2000 PRINT "TONS";
2010 INPUT T
2020 M = T*.9071849
2030 PRINT T;"TONS = ";M;"METRIC TONS"
2040 RETURN
2045 REM ***LINE 2050 IS THE ENTRY OF THE TEMPERATURE CONVERSION SUBROUTINE***
2050 GOSUB 230
2060 PRINT TAB(10);"TEMPERATURE CONVERSION"
2070 PRINT
2080 GOSUB 270
2090 PRINT TAB(5);"1. . . CELSIUS TO FAHRENHEIT"
2100 PRINT TAB(5);"2. . . FAHRENHEIT TO CELSIUS"
2110 INPUT X
2120 IF X <1 THEN 2050
2130 IF X >2 THEN 2050
2140 ON INT(X) GOTO 2150,2200
2150 PRINT "DEGREES CELSIUS";
2160 INPUT C
2170 F = (C*9/5)+32
2180 PRINT CC;"DEGREES CELSIUS = ";F;"DEGREES FAHRENHEIT"
2190 GOTO 2240
2200 PRINT "DEGREES FAHRENHEIT";
2210 INPUT F
2220 C = (F-32)*5/9
2230 PRINT F;"DEGREES FAHRENHEIT = ";C;"DEGREES CELSIUS"
2240 RETURN
9999 END
```

Conclusion

The more we've used the Metric-American Conversion program, the more uses we've found for it. We hope you will enjoy using it, too. As America has already entered the long, painful process of converting to the metric system, I feel certain we will all find a great many uses for it in the future. Incidentally, as you've probably guessed, after *I* had written the program for *her*, Foxy put me to work doing all of the conversions that she had been working on while she went caving. If I live to be 100, I'll just never understand women. ■

```
         METRIC-AMERICAN CONVERSION

DO YOU WANT TO CONVERT:
    1. . . LENGTH
    2. . . AREA
    3. . . VOLUME
    4. . . WEIGHT
    5. . . TEMPERATURE

PLEASE INPUT THE 'NUMBER' OF YOUR CHOICE.
    (INPUT '99' TO END)
? 1

         LENGTH CONVERSION

DO YOU WANT TO CONVERT:
    1. . . KILOMETERS TO MILES
    2. . . METERS TO FEET
    3. . . CENTIMETERS TO INCHES
    4. . . MILES TO KILOMETERS
    5. . . FEET TO METERS
    6. . . INCHES TO CENTIMETERS
? 6
INCHES? 1
1 INCHES = 2.540005 CENTIMETERS

HAVE YOU FINISHED  (YES=1) ? 1
```

Sample run.

The Heath/DEC Connection
(Part 1: Overview of the H11 System)

Familiar to some, unknown to others; Heath's new H11 is certainly a different breed of microcomputer. Find out on the following pages what sets it apart from the rest.

Steven B. Lionel
Kilobaud *Staff*

The story starts back in 1975 when the Altair 8800 made its big splash in the hobbyist computer market. The Heath Company, a recognized leader in electronic kits, decided to get into the burgeoning small-computer field. Heath started to work on their 8080A-based H8, but they wanted something more. They wanted to offer a computer that fit Heath's policy of selling state-of-the-art electronics at a low price. Heath decided to sell a 16-bit computer that could use software already in widespread use.

So it came to pass that Heath, the world's largest electronic kit-maker, joined forces with Digital Equipment Corporation, the world's leading manufacturer of minicomputers. In the fall of 1977, Heath introduced the H11 minicomputer, based on Digital's LSI-11 processor board.

A Heath computer system—the H10 paper-tape reader/punch, the H9 video terminal and the H11 computer. All Heath computer products share a handsome basic cabinet design.

The LSI-11 is the smallest of Digital's -11 line of computers, consisting of the micro-based LSI-11, the older (and still popular) PDP-11 series and Digital's newest addition, the VAX-11. There is a great deal of software available for the -11 computers—much of it usable on an LSI-11. Heath currently offers the H11 with paper tape only, but provides a good selection of languages and utilities.

Current Software

The software that comes with the H11 is all part of Digital's PTS (Paper Tape System) software line. Some of it has been modified by Heath, but every tape carries a Digital copyright. BASIC-11 is an extension of Dartmouth BASIC with excellent string handling and many special functions, as

well as the obligatory PEEK and POKE. Although it is not as powerful as many microcomputer BASICs (no ON . . . GOTO, for example), it is more than adequate for most purposes. BASIC-11 requires 8K words of memory.

A second high-level language included is FOCAL, Digital's own BASIC-like language that is powerful and compact. FOCAL has many features that BASIC-11 does not have, such as formatted output, but has essentially no string capability. There are two versions of FOCAL included with the H11: a 4K version and an 8K version that has such esoteric features as task scheduling and a clock.

The H11 assembler is PAL-11, an easy-to-use two or three pass non-macro assembler. Assembled programs must be processed by the linker, LINK-11S, before they can be run. Also included in the software is ED-11, a good editor that can operate on lines or on characters. It looks much like the other Digital editors and will seem familiar to people who have run on other Digital machines.

ODT-11X is a powerful debugging tool supplied in object form to be linked with other programs. A unique feature of the LSI-11 is that a subset of ODT is a permanent part of the CPU and can be used for debugging without occupying *any* memory at all.

Extremely helpful to assembler programmers is the IOX (Input Output eXecutive) library of I/O subroutines. IOX allows you to write programs without worrying about what I/O device you will be using; this is specified in a single initialization statement. It also provides intermediate editing and buffering on devices. For example, terminal input can be automatically echoed, entire strings can be read (rather than individual characters) and various control-characters, such as rubout/delete and control-U/line-kill, are automatically implemented.

Rounding out the software offerings are two types of memory-dump routines and the absolute loader used to load most programs into memory.

Peripherals

Along with the H11, Heath introduced two low-cost peripherals. The H9 is a video display terminal with a versatile interface and some unique features, including a plot mode and a short-form mode that reformats the screen into 48 twenty-character lines (normal format is twelve 80-character lines). For "mass storage," Heath offers the H10, a 10-character-per-second paper-tape punch combined with a 50-character-per-second paper-tape reader.

All Heath computers and peripherals share a handsome basic enclosure design, as shown in the photos. Heath also offers Digital's best-selling LA36 DECwriter II, a 30 cps matrix-printing terminal. The H36, as Heath calls it, is sold assembled at a price that is nearly impossible to beat for single quantities.

Future Offerings

Heath has made a major investment in their computer line—hiring many engineers who are racing to get new products on the market. The primary item missing from the current H11 system is a random-access storage device.

Heath's cassette for the H8 is not compatible with the H11; however, Heath is working on a floppy-disk drive and will release it toward the end of the year. The drive will be by Memorex and have a controller based on Western Digital's floppy-disk controller chip. It will have a 256K byte capacity with soft sectoring, and will be available in both single- and dual-disk configurations.

Heath will use its own format for writing on the disk, but will provide a switch that will allow it to write in a format compatible with Digital's RXV11 floppy-disk system. This means you could run Digital's RT-11 operating system on the H11, although the software license, at $2760, will probably set you back a good deal. Instead, Heath is developing its own HDOS operating system. Similar to RT-11, it is a cooperative effort between Heath and Digital. HDOS will come with a disk BASIC and FOCAL, along with supporting software.

The future holds a lot for H11 owners. According to Heath, a "major software effort" is being expended to produce FORTRAN, APL and PASCAL for the H11. On the hardware side, Heath plans to sell Digital's LSI-11/2 processor, which is half the size of the current LSI-11 and does not include the on-board 4K words of RAM. Heath may also offer Digital's higher-density RAM boards of 8K, 16K and 28K words, making possible a system with the maximum amount of RAM (56K bytes), a floppy, a serial and a parallel interface with a slot left over, all inside the H11's cabinet.

The Heath H11 computer. Digital's LSI-11 processor is what makes it go.

Computer and Options

The H11 microcomputer consists of an assembled and tested Digital KD11-F LSI-11 processor board, an eight-slot backplane and card cage, and a regulated power supply. The power supply can be set by a switch to run from either a 120 V or 240 V line voltage. The voltages are regulated before being distributed to the backplane, so that no heat-producing on-board regulators are needed.

There are only two switches on the front panel. One controls the dc power on the backplane and the other is the RUN/HALT control. When the switch is in the HALT position, the H11 can single-step program execution. Missing from the H11 is a switch controlling the line time clock (one controlled by line frequency), as is found on the PDP-11/03. Instead, this function is controlled by a jumper on the power-supply board.

The H11, essentially a PDP-11/03 in kit form, is completely compatible with all existing LSI-11 modules. Therefore, the bus used is Digital's Q-BUS, a 36-line, 16-bit bus that has built-in interrupt priority determination and allows DMA (direct memory access), where I/O can take place without the processor's services.

As nice as the Q-BUS is, it has a negative side for hobbyists: Since Digital jealously guards its patents on the -11 computers, no one can sell anything that plugs into the Q-BUS without a license from Digital. This will probably prevent the widespread availability of inexpensive modules for the H11.

The LSI-11's system architecture is radically different from the common 8-bit microprocessors. First, there is no accumulator. Instead, there are eight high-speed registers, any of which may be used as source or destination for most instructions. Even more unusual is that memory locations can be used as operands with equal ease. This allows arithmetic, comparison and program control without ever referring to registers. Additional addressing modes are register indexed,

Glossary of Jargon

String Handling: This does not, as it sounds, have anything to do with a cat's-cradle exchange. In computer jargon a string is any series of characters treated as a group (for example, a person's name would be a string of letters, an address would be a mixture of letters and numbers, etc.). Some computer languages are better than others at dealing with these groups of characters. This is called string handling.

Powerful: This is a beauty. *It has no real meaning.* In advertising literature it means the advertised system is better than others, but usually for undefined reasons. The power of a computer is its ability to do things, so a more powerful computer can do more in a given time. In general, the term is used when a writer feels intuitively that a system is better than others, but is unable to point out the reasons.

Formatted Output: Let's say you want to print a name, address, city, state and zip code for mailing. You'll want this to be in three lines so the post office can (you hope) read it. To do this you have to have a format program that tells the computer to put the name on the first line, etc. If you wanted the information all on one line for a 132-column computer page printout, you would format the printing differently. Some languages make it easy to set up these formats; others require all sorts of programming work.

Task Scheduling: The LSI-11 system is set up so it can seemingly do two things at once. Let's say you want to sort out a list of names and addresses into zip-code order from alphabetical order by last name. This is going to take a while. In the meantime, while the system is grunting away at the sorting program, you can play a game of Star Trek. The H11 will work on the sort without bothering you. When you key in a command it stops the sorting program, files away what it was doing so it can go back to it after it has attended to your demands, then goes to your Star Trek program and executes what you've keyed in. When that is done, a few microseconds later, it puts all your stuff in another file and gets out the sorting program and continues with that. This is called task scheduling.

ON-GOTO: As you become more familiar with BASIC you'll understand about this. Until then don't worry about it. The ON-GOTO is a time-saver in programming, but you can make do without it by adding a few extra programming steps.

Editor: The editor of a magazine corrects spelling, blue-pencils words, etc. The computer editor makes it possible to do these things. The better the editor, the more things it can do. A simple editor might enable you to delete a character on your tube . . . a word . . . a line. It can allow you to move the cursor around where you wish. It might permit you to add a character or word into a line. A more involved editor could allow you to search a string of words for any particular word or pattern of words you wanted. The more functions an editor has, the more "powerful" it is.

Clock: The word "clock" is used in computers to mean several things. Every computer has to have a time-generating system to step it through its operations. This is called a clock. The speed of this clock determines the speed of a computer. Most of the 8080-type systems run at around 2 MHz . . . the 6800 at around 750 kHz, the 6502 at around 1 MHz and the Z-80 at 3 MHz, generally. The H11 runs at 10 MHz. Another type of clock, usually called a "real-time clock" to identify it from the "system clock," keeps track of time for you. The PET comes with this clock built in so all you have to do is set it when you plug in the computer and it will keep reasonably good time for you. The Altair-type systems require that an accessory real-time clock board be used—this usually costs around $100 or a little less. This is the type of clock referred to as being built into the H11.

Wayne Green

indirect, absolute, autoincrement and autodecrement (useful for stacks) and relative mode for branching.

The LSI-11 implements the 400+ instruction set of the PDP-11/40, although not as quickly. The instruction set is well designed and efficient, with many instructions operating either on bytes or on words. Available as an option is a special ROM chip that implements 32-bit integer and floating-point arithmetic as found on the PDP-11/45.

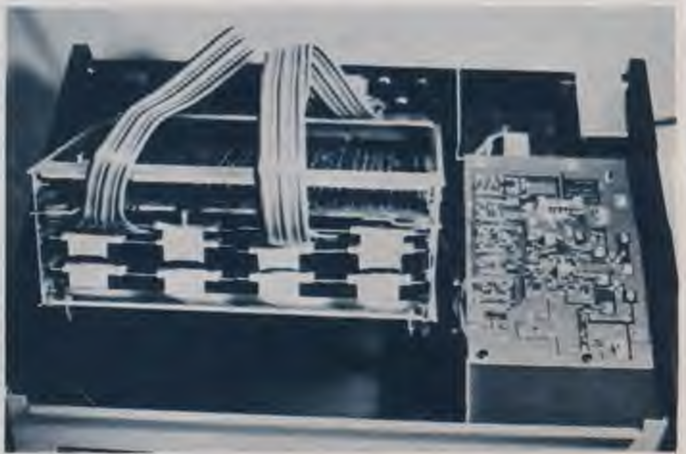
The KD11-F processor board contains the four processor chips, a socket for the optional EIS/FIS ROM chip described above, all necessary logic to control the Q-BUS and 4K of dynamic RAM. The on-board RAM can be used as the 0-4K area or the 4-8K area, depending on a jumper. Optionally, the on-board RAM may be completely disabled, allowing other memory to occupy the low addresses. The processor board generates a refresh signal for dynamic memories, although the current offering by Heath is static.

Because of its size, 8.5 x 10 inches, the KD11-F board occupies two backplane slots, although only one is used. If the new KD11-HA LSI-11/2 processor-only board is used, the extra option slot will be free. Heath plans to offer this new board soon.

At this time, Heath offers three accessory boards for the H11. The H11-1 is a 4K-word static-RAM memory board with an average access time of 500 ns. The H11-1 uses 2114 1K-by-4-bit memory chips and can be jumpered for any of the eight memory banks. The H11-2 parallel interface module allows easy control of parallel devices such as the H10 paper-tape reader/punch. The third offering is the H11-5 serial interface, which can be jumpered for various baud rates and signal types. Although all H11 option boards are Heath designed, all are completely compatible and interchangeable with existing LSI-11 boards.

Building the H11 is a breeze. The CPU board is already assembled and tested. One circuit board is used for the power supply and one for the backplane. Heath's use of solder-masked boards makes solder bridges nearly impossible.

With their new computer line, Heath surpassed themselves on documentation. The H11 comes with a padded vinyl binder stuffed with hundreds of pages of information about the instruction set, Q-BUS, logic descriptions, troubleshooting, schematics and hundreds more on the software. However, Heath's technical writers have not been quite so prolific, since much material is reproduced from Digital publications. Still, there is no lack of information on the H11, and it is welcome.



Inside the H11 with backplane raised. Ribbon cables connect peripheral controllers to sockets on the back panel. Power supply is on the right.

My only complaint is that the binder supplied simply won't hold all the pages in place!

H11—Is It Worth It?

There is a question floating around that is usually phrased as, "LSI-11 vs 8080, it is worth twice the price?" The answer depends on what you are looking for. If you want your computer to sing the Star Trek theme while you zap the Klingons in multicolor graphics, the H11 is probably not for you. However, if you want serious computing capabilities for professional, small business or educational environments, the H11, at $1295, will give you performance that no 8-bit microcomputer can touch. The H11 is no plaything, and in my opinion, is the best microcomputer on the market.

Next month we'll discuss the H11's peripherals. ■

The following are trademarks of the Digital Equipment Corporation: LSI-11, PDP, DIGITAL, FOCAL, DECUS, Q-BUS, VAX, DECwriter.

BASIC is a trademark of the Trustees of Dartmouth College.

Teletype is a trademark of the Teletype Corporation.

Heath is a trademark of the Heath Company.

Clockwise from top—LSI-11 CPU board, H11-5 serial interface, H11-2 parallel interface and H11-1 4K RAM.

Home System Demo Program

A demonstration program is valuable for hobbyists showing off their micros. It's more valuable for retailers trying to sell systems to customers who don't know computers.

Mark Herro
311 Woodland Ln
Oconomowoc WI 53066

I'll level with you. I don't have a computer (yet). The software you may have seen from me on the pages of *73* and *Kilobaud* came from either a time-sharing service's system or was mooched from somebody else's microcomputer. Don't get me wrong—those *are* my programs; I just "borrowed" a computer to debug and run them.

Introduction . . . The Question

Even though I'm not a true "hobbyist" yet, I've learned a lot from *Kilobaud* and other sources—particularly that a lot of the software out there is merely *games*, sophisticated or otherwise. I'm not saying games are bad, but how does the abundance of them appear to the complete outsider who thinks computers are just nasty machines that foul up credit bills? "Just an expensive toy," they might think, leading to that often heard and "loved" question: "What else can you do with it?" "Anything you can think of!" is not a sufficient answer!

The Answer

I'm aiming this article at two groups of people. One is computer users (with up and running machines) who want to show off their gear to noncomputer types without scaring them away with the buzzwords or by playing games all night.

The other group is *computer-store retailers!* Refer to John Craig's editorial in *Kilobaud* No. 6. If John had really been the people he pretended to be, I think that he would have been turned off to computers for

I. Introduction
 A)"I'm Mark's computer . . . "
 B)Brief history (" . . . Mark built me about a year ago . . . ")
 C)Overview of DP (" . . . I'm going to show you . . . ")
II. Small Talk
 A)Name query
 B)Age query
 C)Comment on name/age
III. Demonstration of security system
IV. Educational program demonstration
V. Demonstration of graphics display
VI. Home accounting/budget program
VII. Demonstration of ham radio system
VIII. Demonstration of programming capabilities
 A)Short explanation of BASIC
 B)Math programs
 C)English programs
 D)Other possibilities
IX. Conclusion

Fig. 1. Possible DP outline.

quite a while.

What is really needed is something to show off the capabilities of a computer system that isn't too technical, and doesn't play games: a Demonstration Program—I'll call it a DP for short—with brief demonstrations of a computer's abilities. A DP would probably have to be different for each computer (i.e., Mits, SWTP, Poly, Imsai, Compucolor, etc.) and each computer's available memory, operating systems, I/O, special functions and the like. In other words, a DP would almost have to be tailor-made to each and every computer running. Think of what could be done. The individual programming possibilities are almost endless! As a starting point for a DP, I will assume the following conditions: (1) you have a computer; (2) you have a high-level language; (3) you want to show off your computer; (4) you want to sell your computer.

What Should Be In It?

What should a DP contain? Take a look at your computer's *outside* hardware and directly observable equipment (*not* internal processor timing, bus structure, DMA, or other such "buzzwordy" nonsense. You may say "Waddya mean, 'nonsense?!'"—it's all GIGO (Garbage In/Garbage Out) to people not familiar with computers.). Look for things like graphics modes. Can the video readout run on a color TV (like Apple-II and Compucolor)? Can the computer play (or compose) music? Is it used for home security? Ham radio? Anything that has a direct outside-world link can be used in a DP.

Even if you don't have any of the above capabilities, you can still run a "bare-bones" DP. A good place to get ideas for this "no frills" type of DP might be the "Hello" program (game) in *101 Computer Games*. In this program, a person may have a "conversation" with the computer (which acts as a sort of psuedo problem-solver) and get appropriate answers to user questions.

A point should be taken from this: A DP should be *interactive* between the "demonstrator" and the "demonstratee." Initially, a DP could ask for names and ages, and, if possible, engage in some small talk. The ideal DP would show off all the computer's capabilities while at the same time interacting with the user. Fig. 1 shows a sample DP outline.

Of course, all the demonstrations would only be as long as the programmer wanted to or could make them; I must emphasize that there wouldn't be just one type of DP. A DP would have to be tailored to each system's capabilities. A sample of a partial DP printout is shown in Fig. 2.

Conclusion

A DP would be invaluable to hackers trying to show off their microcomputers. It would be even more valuable to the retailer (manufacturer?) who must try to *sell* his systems to non-computer-oriented customers. I hope I have given some good ideas to someone out there. Get going.■

```
                ########COMPUTER DEMONSTRATION#######
HELLO! I'M MARK'S COMPUTER. I BET YOU THINK COMPUTERS
ONLY MESS UP YOUR BILLS, RIGHT?
                                  WRONG!!
I AM GOING TO SHOW YOU SOME OF THE THINGS MARK TAUGHT ME TO
DO. MARK ONLY BUILT ME A YEAR AGO, BUT I CAN DO ALL SORTS OF
NEAT THINGS.
BUT FIRST, I WOULD LIKE TO KNOW WHO I AM TALKING (WELL,
PRINTING) TO. WHAT'S YOUR NAME? BILL
HELLO BILL, HOW OLD ARE YOU? 205
A COMEDIAN, HUH?
OK, MOSES, HOW OLD ARE YOU—REALLY? 25
ALMOST OVER THE HILL, COMPARED TO ME.
OK, BILL, NOW I WILL SHOW YOU WHY A BURGLAR SHOULDN'T COME
TO THIS HOUSE AND TRY TO STEAL ME.
SEE THAT WINDOW JUST TO THE RIGHT OF ME? YES
OPEN IT.

                       (The window is opened and
                        the bell on the terminal
                        goes off)

OK, SHUT THE WINDOW! SHUT THE WINDOW!
IF THIS WASN'T A TEST, I WOULD HAVE SOUNDED THE WHOLE HOUSE
ALARM SYSTEM, TURNED ON THE LIGHTS, AND IF MARK DIDN'T GIVE
ME A SPECIAL COMMAND, I WOULD CALL THE POLICE

                                NEAT, HUH?

BILL, ARE YOU A GOOD ARTIST? NOT REALLY
OH, BILL, I AM STILL PRETTY DUMB WHEN IT COMES TO LANGUAGE.
JUST ANSWER YES OR NO.
? NO
WELL, YOU CAN HAVE ME DRAW ALL SORTS OF PICTURES (EVEN DIRTY
ONES). BUT JUST TO DEMONSTRATE, I'LL WRITE YOUR NAME IN
BIG LETTERS . . .

BBBBBBBB     IIIIIIIIIIII     LLLL             LLLL
BBB     BB      IIIIII        LLLL             LLLL
BBB     BB      IIIIII        LLLL             LLLL
BBB    BBB      IIIIII        LLLL             LLLL
BBB     BB      IIIIII        LLLLLLL          LLLLLLL
BBB     BB      IIIIII        LLLLLLLLL        LLLLLLLLL
BBBBBBBB     IIIIIIIIIIII     LLLLLLLLLLL      LLLLLLLLLLL

I AM PRETTY GOOD AT PLOTTING MATHEMATICAL CURVES TOO

                  (rest of the program runs through)

SO YOU SEE BILL, YOU CAN USE ME FOR ALL SORTS OF THINGS; IT'S
JUST A MATTER OF CORRECT PROGRAMMING. IT ISN'T HARD, ONCE
YOU GET THE HANG OF IT.
NOW THAT YOU KNOW MORE ABOUT ME, GO AHEAD AND PLAY SOME GAMES.
TAKE THE CASSETTE MARKED "GAMES" AND PUT IT INTO THE RECORDER.
YOU HAVE A WHOLE BUNCH TO CHOOSE FROM, BUT I RECOMMEND "TREK"
OR "DEPTH CHARGE"—HAVE FUN!! BYE.
READY
```

Fig. 2. Partial DP sample printout.

Do-It-All Expansion Board for KIM

The following design and repackaging information should appeal to many, if not all, the KIM-1 owners out there who are looking for more in this very popular system.

George Young
Sierra High School
Tollhouse CA 93667

Breathes there a KIM-1 owner who is completely satisfied with his KIM-1? The number of articles appearing about this very popular computer and about repackaging it surely mean that many owners seek a more presentable microcomputer with expanded capabilities. This article presents an expansion board for the KIM-1 as well as some ideas for repackaging the basic KIM-1 circuit board.

The Expansion Board

The expansion board is the same size as the KIM-1 circuit board and is designed to mount below the KIM-1 board with stand-offs. The board is a "universal type," meaning that it will take 14-, 16-, 24- and 40-pin ICs and that it can be either wire-wrapped or hard-wired. I plan to use my board to add: (1) more memory, (2) Don Lancaster's TVT-6L circuitry from *Kilobaud* No. 6, (3) Bob Grater's SAB-1 circuitry from *Kilobaud* No. 1, (4) a Selectric interface similar to the one described in "another magazine," (5) whatever else is published for KIM that I need and can use.

Photo 1. Expansion power supply.

Since the expansion board is almost the same size as a page of *Kilobaud*, some reduction of the artwork is required to fit it into the magazine. If you are going to home-brew your own board, you will need to use the skills and techniques presented in "Make Your Own PC Boards" (*Kilobaud* No. 16, page 24).

In trying to anticipate your needs, we have made arrangements for the Byte Shop, 3139 E. McKinley Ave., Fresno CA 93703, to produce the board for you at $35 postpaid. It will be a single-sided, G-10 glass-epoxy board with numerically controlled drilled holes. (You couldn't drill all the holes on that artwork for $35 worth of labor!) Mating connectors for the board are also available from the same source for $5 per pair.

Should you find it necessary to have contact fingers available on both sides of the board, there are two ways to accomplish this.

The first way is the easiest for those who live close enough to an Electronic Supermarket. Obtain Circuit Stik™ part number 3396-002; you will need just one package per KIM-1 expansion board. This 22-pin printed-circuit-board edge connector will mate with the sockets and comes two per package. (Just be sure you get them anchored at the correct location on the back of the KIM-1 board.)

Should you be "out of range" of the Electronic Supermarket, the task will be a little more difficult. You'll have to home-brew a device like the Circuit Stik edge connectors. There is a flexible copper-clad epoxy board available that you can print on in the fashion described in "Make Your Own PC Boards." The edge connectors formed in this fashion can be glued to the back of the KIM-1 board. I use Goodyear Pliobond™ general-purpose adhesive for this purpose.

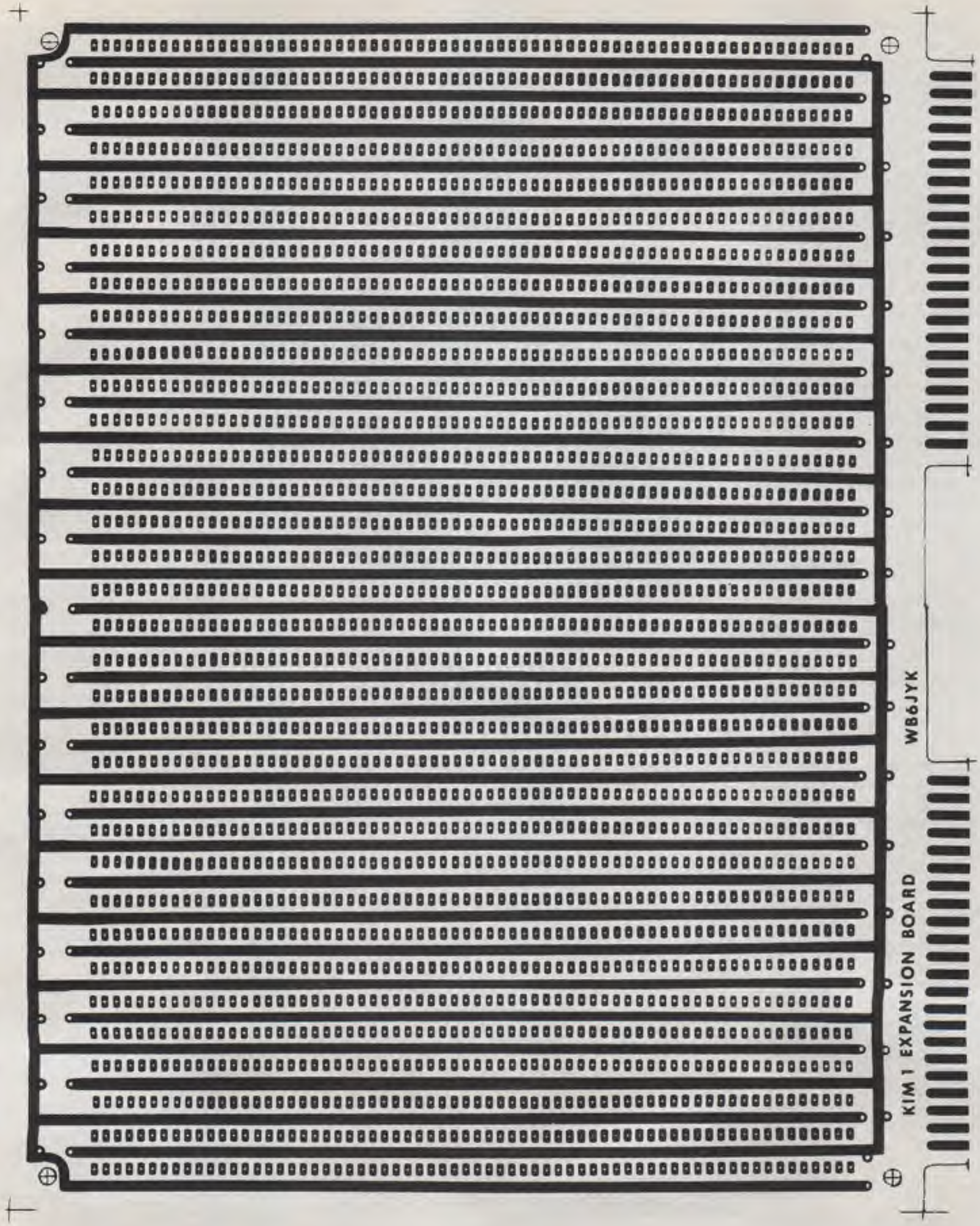


Fig. 2. Original artwork for the KIM-1 expansion board (reproduced here at 85 percent).

Extra-thin copper clad is also available in the Circuit Stik line as part number 9252 (3 × 4 inches) and part number 9251 (5 × 6 inches). The Circuit Stik material comes with a self adhesive backing, so use of this material bypasses the gluing problem. If you cannot find it locally, then order from Circuit Stik, 24015 Garnier Street, PO Box 3396, Torrance, CA 90510, or call (213) 530-5530 and ask where you can buy their products.

Photo 2. The wooden "briefcase."

Photo 3. KIM's new "home," ready to travel.

KIM-1 Power Expansion

The first thing that will be needed for the expansion circuitry is additional power capability. Fig. 1 shows my version of the new KIM-1 power supply. Photo 1 also shows the new power supply. The transformer used is a rewound core from the vacuum-tube era and has a capacity of about 100 Watts. Details of the power-supply theory of operation have been covered in the Kilobaud Klassroom series, so I shouldn't have to repeat it here. (How else will I get you to read my other stuff?)

Photo 4. Elevating the keyboard and the display.

A New Enclosure

After studying the various ways that others have given KIM a new housing, I decided to do mine the easiest way possible. The problem is that those capacitors on the KIM circuit board stick up too high. What can be done is to raise the keyboard and the display about ½ inch so a panel will clear these capacitors.

Using a temperature-controlled iron and a vacuum desoldering device, I carefully desoldered the keyboard. I stripped hookup wire and soldered pieces about 1 inch long to the keyboard contacts. I placed a ½-inch-thick block of wood under the keyboard and reconnected the hookup wire to the KIM-1 circuit board. Then I carefully checked out KIM for normal operation.

The display also had to be elevated about ½ inch; so I carefully desoldered the display module and then cleaned solder from the seven-segment readout pins. I removed pins from an Augat™ wire-wrap module and slipped each pin over one of the pins on the LED readouts. Finally, I soldered the Augat pins to the KIM circuit board. This raised the display (see photo). Again, I carefully checked KIM for normal operation.

Everything looked OK, so I made a cardboard template that would exactly fit over the KIM keyboard and the display module, and then used this to make the cutouts in a 1/16-inch-thick piece of plastic laminate.

A briefcase-style case made of walnut-stained wood formed the new home for KIM (a simpler method would be to use an actual briefcase). Photos 2 and 3 show the completed KIM-1 package. I plan to add connectors, probably down the middle of the plastic laminate, for connection of the various peripherals. In the meantime, KIM-1 is now well protected and is portable. My students can (and are) packing it home almost every night so they can use KIM in its more-or-less original configuration.

A Final Note

The 1/16th inch plastic laminate is too flexible and bends when the keyboard is operated. To overcome this, I glued wooden strips to the back of the plastic laminate with a hot-glue gun to give the panel sufficient rigidity. ■

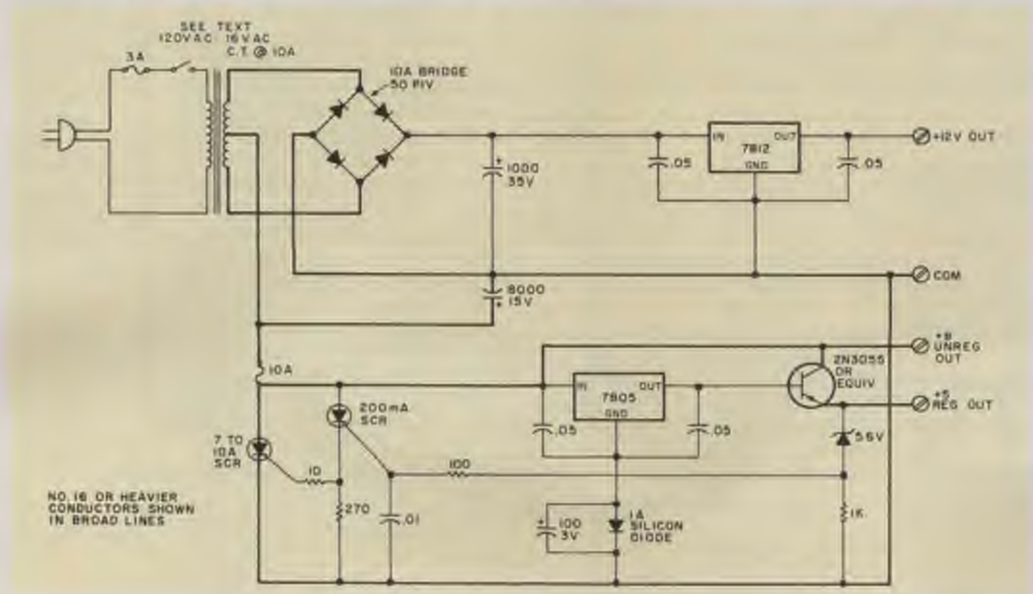


Fig. 1. KIM-1 expansion power supply.

Jack A. Inman
922 E. Edgecomb Ave.
Covina CA 91724

Tally Ho!

This strategic game program offers proof that fox hunting is not obsolete.

Fox and Hounds is an old game — a variation of checkers played on a standard checkerboard. Unlike checkers, this game is played with only five pieces on the board. There are one fox and four hounds; one player moves the fox, the other the hounds. In this case the computer moves the "fox," and the human player moves the "hounds."

Since the number of pieces is limited, neither side can jump pieces. The hounds can move in a forward direction only. The solitary fox would run out of moves very quickly if allowed movement only in one direction, so he can move either forward or backward. All moves for both fox and hounds are limited to the standard checker move — that is, one space diagonally per move.

The object of the game is for the fox to reach the safety of line one (1). If the fox can get to line one, the computer (fox) wins. If the human player can trap Renard in a location from which he cannot escape, the human wins.

I wrote this game in SWTP 8K version 2.0 BASIC, which supports multiple statements on each line. However, since most BASICs do not have this feature, I chose not to use it.

I also tried to use only the standard BASIC statements that should be available in any good version. I did this so anyone can type the program into his machine without having to rewrite half the program. (I hate to have to try to figure out what some special statement is supposed to do.)

I did use one nonstandard statement, PRINT CHR$(16); CHR$(22), several times to cause my terminal (SWTP CT-1024) to home up and clear the screen. If you have this feature, fine; no change is required. If you do not, you can omit the lines and the program will work anyway. If you have scrolling, simply omit these lines.

Program Description

The program listing shows the program starting in a standard game fashion: asking if the player wants instructions. If the player's answer is yes, the instructions are printed out. Even though the program prints TYPE YES OR NO, anything other than a yes answer is ignored, causing the program to go on to the next line. I see no good reason to force the player to type NO if he does not want the instructions.

The board array is set by the DIM statement in line 110. The array is set to 9 by 9, even though the board is only 8 by 8, so the automatic moves by the computer do not test out of bounds of the array and cause error stops.

Line 330 sets up the fox in his initial position. Lines 340 to 370 set up variables to keep track of the fox's position for the automatic moves. They are also used to print

Program listing.

```
0010 REM *** FOX AND HOUNDS ***
0020 REM *** A CHECKER GAME ***
0030 REM *** PROGRAM BY JACK A. INMAN ***
0040 REM *** OCTOBER 1977 WRITTEN IN SWTPC 8K BASIC ***
0050 PRINT "WANT INSTRUCTIONS";
0055 PRINT "TYPE YES OR NO";
0060 INPUT A$
0070 IF A$="YES" GOSUB 1500
0100 REM *** SET LIMITS ***
0110 DIM L(9,9)
0200 REM *** GENERATE BOARD ***
0205 REM *** GENERATE LINE 1 ***
0210 FOR C=1 TO 8 STEP 2
0220 L(1,C)=1
0230 L(1,C+1)=2
0240 NEXT C
0245 REM *** GENERATE LINES 2 THROUGH 8 ***
0250 FOR C=1 TO 8 STEP 2
0255 L(2,C)=4
0260 L(2,C+1)=1
0265 L(3,C+1)=4
0270 L(3,C)=1
0275 L(4,C)=4
0280 L(4,C+1)=1
0285 L(5,C)=1
0290 L(5,C+1)=4
0295 L(6,C)=4
0300 L(6,C+1)=1
0305 L(7,C)=1
0310 L(7,C+1)=4
0312 L(8,C)=4
0315 L(8,C+1)=1
0320 NEXT C
0325 REM *** SET UP FOX INITIAL POSITION ***
0330 L(8,5)=3
```

out the computer's moves for the player.

The screen is cleared and homed by line 405. Next, the board is printed out by lines 410 through 530. Line 406 branches to the subroutine that prints the computer's moves for the player. During printing of the board, the letters X, H, F and O are printed, dependent upon the value in the array position. The X indicates a square that cannot be moved onto; the H is the position of a hound; the F is the fox's position; and the 0 is a vacant square that can be occupied by a piece.

I used the designations F and H for the fox and hounds rather than the conventional R (red) and B (black) because this made the pieces easier to track.

The board is printed out after each move. After each printout, lines 540 and 560 test for a win by either the computer or the human.

Lines 600 through 640 obtain the player's move. Each move requires two coordinates to tell which hound is to be moved. Two more coordinates are required to tell where the player wants to move the hound. The required format is line first, then column, for all coordinates.

The validity of the move is tested by lines 650 through 710. Valid moves can be only one space forward and one diagonally, right or left. The space the player wants to move to must also be vacant (indicated by the presence of a 4 in the position of the array). The screening also verifies that the player is attempting to move only a hound, which is indicated by a 2 in the position of the array, the position being the "from" (F,F1) coordinates for the hound test. The "to" test is made on the T,T1 coordinates.

If the tests prove the move is valid, lines 720 and 730 reverse the position of the hound and the vacant square in the array. Thus the hound is moved.

The computer moves next. Lines 903 through 940 randomize the left-right diagonal movement of the fox. Without this, movement would always be either diagonally left or right, depending upon which came first in the program, and the fox's moves would be too predictable. The Q variable is used to assure that all possible moves are tested. Forward moves are always tested before backward moves are attempted. (Remember, the fox is trying to get to line one.)

Lines 950 through 1270 do the actual testing and moving for the fox. This is one of the places the variables X, Y, A and B are used to keep track of the fox's position. In each of the four possible moves, the position to be attained is tested for vacancy. If it's vacant, the fox is moved.

Next, the space the fox was in is changed to a vacant space (4). If no vacant space is found for the move being tested, the value of Q tells the program which move to try next until all move possibilities are exhausted. If no more moves are available, the variable Z will be set to 1 by line 1170 or 1260, indicating to line 560 in the board printout that the human has won the game. It also causes a computer-move printout of I CAN'T MOVE.

If, however, the computer can find a valid move, the values of X and Y are left holding the position in the array for the moved-to position. The values of A and B retain the moved-from position in the array. All of these are used later to print out the computer's move.

Playing the Game

A run of the program is included. It shows what the board looks like and how the computer requests moves from the player. Each line is numbered for convenience; each column is numbered below the column. The com-

```
0335 REM *** LOCATION HOLDERS ***
0340 X=8
0350 Y=5
0360 A=X
0370 B=Y
0400 REM *** CLEAR SCREEN AND PRINT BOARD ***
0405 PRINT CHR$(16);CHR$(22)
0406 GOSUB 1400
0410 FOR L=1 TO 8
0420 FOR C=1 TO 8
0430 IF L(L,C)=1 PRINT"X ";
0440 IF L(L,C)=2 PRINT"H ";
0450 IF L(L,C)=3 PRINT"F ";
0460 IF L(L,C)=4 PRINT"O ";
0470 NEXT C
0480 PRINT " L";L
0490 NEXT L
0500 FOR K=1 TO 8
0510 PRINT K;
0520 NEXT K
0530 PRINT
0535 REM *** TEST FOR COMPUTER WIN ***
0540 IF X=1 GOTO 1350
0550 REM *** TEST FOR HUMAN WIN ***
0560 IF Z=1 GOTO 1300
0600 REM *** HUMAN MOVES ***
0610 PRINT "YOUR MOVE FROM(LINE,COLUMN) ";
0620 INPUT F,F1
0630 PRINT "TO ";
0640 INPUT T,T1
0645 REM *** TEST FOR VALID MOVE **
0650 IF T <F PRINT"YOU CAN'T MOVE BACKWARDS"
0655 IF T <F GOTO 600
0660 IF T-F <>1 GOTO 700
0680 IF L(F,F1) <> 2 GOTO 700
0690 IF L(T,T1)=4 IF L(F,F1)=2 GOTO 720
0700 PRINT "INVALID MOVE"
0710 GOTO 600
0715 REM *** MOVE IS VALID--MOVE PLAYER ***
0720 L(T,T1)=2
0730 L(F,F1)=4
0900 REM *** COMPUTER MOVES ****
0903 Q=0
0905 K=INT(RND(0)*2)
0910 IF K >2 GOTO 905
0930 IF K=1 GOTO 950
0940 IF K=0 GOTO 1060
0945 REM *** TEST (1) FORWARD:(1) LEFT ***
0950 Q=Q+1
0953 X=A-1
0955 Y=B-1
0960 IF L(X,Y)=4 L(X,Y)=3
0970 IF L(X,Y) <>3 IF Q <>2 GOTO 1060
0975 IF L(X,Y) <>3 IF Q=2 GOTO 1100
0980 IF L(X,Y)=3 L(X+1,Y+1)=4
0990 IF L(X,Y)=3 GOTO 400
1050 REM *** TEST (1) FORWARD (1) RIGHT ***
1060 X=A-1
1062 Y=B+1
1065 Q=Q+1
1070 IF L(X,Y)=4 L(X,Y)=3
1080 IF L(X,Y) <>3 IF Q <2 GOTO 950
1082 IF L(X,Y) <>3 IF Q=2 GOTO 1100
1085 IF L(X,Y)=3 L(X+1,Y-1)=4
1090 IF L(X,Y)=3 GOTO 400
1100 REM *** COMPUTER MOVES BACKWARDS ***
1101 IF K=0 GOTO 1105
1102 IF K=1 GOTO 1200
1104 REM *** TEST (1) BACK (1) RIGHT ***
1105 Q=Q+1
1110 X=A+1
1115 Y=B+1
1120 IF L(X,Y)=4 L(X,Y)=3
1130 IF L(X,Y) <>3 IF Q < 4 GOTO 1200
1150 IF L(X,Y)=3 L(X-1,Y-1)=4
1160 IF L(X,Y)=3 GOTO 400
1170 IF L(X,Y) <>3 Z=1
1180 GOTO 400
1200 REM *** TEST (1)BACK (1) LEFT ***
1210 Y=B-1
1211 X=A+1
1215 Q=Q+1
1220 IF L(X,Y)=4 L(X,Y)=3
1230 IF L(X,Y) <>3 IF Q <4 GOTO 1105
1240 IF L(X,Y)=3 L(X-1,Y+1)=4
1250 IF L(X,Y)=3 GOTO 400
1260 IF L(X,Y) <>3 Z=1
1270 GOTO 400
1300 REM *** HUMAN WINS ***
1310 PRINT "YOU WIN . . . WANT TO PLAY AGAIN ";
```

puter moves are printed above the board; the request for the player's move is printed below the board. The request for the move contains a reminder to type line and then column in that order.

The printout of the board shows all of the vacant squares as represented by the letter O. Nonvalid squares are represented by the letter X. The player can move onto any vacant square in the line which is one line greater than the line and one column diagonally from the current position of the hound to be moved (standard checker move). I did not include any nonvalid moves in the run; however, the screening does work.

That about covers the game. Good luck to the hounds! ■

```
1320 GOTO 1370
1350 REM *** COMPUTER WINS ***
1360 PRINT "I WIN . . . WANT TO PLAY AGAIN ";
1370 INPUT A$
1380 IF A$="YES" GOTO 200
1390 END
1400 REM *** PRINT COMPUTER MOVE ***
1405 IF A=X GOTO 1440
1406 IF Z=1 PRINT"I CAN'T MOVE"
1407 IF Z=1 GOTO 1440
1410 PRINT "I MOVED FROM ";A;" ";B;"TO ";X;" ";Y
1420 A=X
1430 B=Y
1440 RETURN
1500 REM *** INSTRUCTIONS ***
1505 PRINT CHR$(16);CHR$(22)
1510 PRINT "YOU ARE THE HOUNDS."
1520 PRINT "YOU HAVE 4 PLAYERS (H)."
1530 PRINT "YOU CAN MOVE FORWARD ONLY."
1535 PRINT
1540 PRINT "THE COMPUTER IS THE FOX (F)."
1550 PRINT "THE FOX CAN MOVE BACK AND FORWARD."
1555 PRINT
1560 PRINT "IF YOU TRAP THE FOX YOU WIN."
1570 PRINT "IF THE FOX GETS TO LINE 1"
1580 PRINT "THE COMPUTER WINS."
1590 PRINT "TYPE A RETURN TO CONTINUE";
1600 INPUT A$
1610 PRINT CHR$(16);CHR$(22)
1620 PRINT "TO PLAY. .WHEN IT IS YOUR MOVE"
1630 PRINT "TYPE IN THE LINE NUMBER"
1640 PRINT "COMMA COLUMN NUMBER OF THE"
1650 PRINT "MAN YOU WANT TO MOVE."
1655 PRINT
1660 PRINT "THEN WHEN I ASK TO?"
1670 PRINT "TYPE IN LINE NUMBER"
1680 PRINT "COMMA COLUMN YOU WANT TO"
1690 PRINT "MOVE TO. GOOD LUCK"
1700 PRINT "TYPE A RETURN WHEN READY";
1710 INPUT A$
1720 RETURN
```

```
RUN

DO YOU WANT INSTRUCTIONS
TYPE YES OR NO ? YES

YOU ARE THE HOUNDS.
YOU HAVE 4 PLAYERS (H).
YOU CAN MOVE FORWARD ONLY.

THE COMPUTER IS THE FOX (F).
THE FOX CAN MOVE BACK AND FORWARD.

IF YOU TRAP THE FOX YOU WIN.
IF THE FOX GETS TO LINE 1
THE COMPUTER WINS.
TYPE A RETURN TO CONTINUE?

TO PLAY..WHEN IT IS YOUR MOVE
TYPE IN THE LINE NUMBER
COMMA COLUMN NUMBER OF THE
MAN YOU WANT TO MOVE.

THEN WHEN I ASK TO?
TYPE IN LINE NUMBER
COMMA COLUMN YOU WANT TO
MOVE TO. GOOD LUCK
TYPE A RETURN WHEN READY?

X H X H X H X H  L1
O X O X O X O X  L2
X O X O X O X O  L3
O X O X O X O X  L4
X O X O X O X O  L5
O X O X O X O X  L6
X O X O X O X O  L7
O X O X F X O X  L8
1 2 3 4 5 6 7 8
YOUR MOVE FROM(LINE, COLUMN) ? 1,4
TO ? 2,3

I  MOVED  FROM  8  5  TO 7  4
X H X O X H X H  L1
O X H X O X O X  L2
X O X O X O X O  L3
O X O X O X O X  L4
X O X O X O X O  L5
O X O X O X O X  L6
X O X F X O X O  L7
O X O X O X O X  L8
1 2 3 4 5 6 7 8
YOUR MOVE FROM(LINE, COLUMN) ? 1,2
TO ? 2,1
```

(Program continues until . . .)

```
I  MOVED  FROM  5  2  TO  4  1
X O X O X O X O  L1
H X O X O X O X  L2
X H X O X O X O  L3
F X H X O X O X  L4
X O X H X O X O  L5
O X O X O X O X  L6
X O X O X O X O  L7
O X O X O X O X  L8
1 2 3 4 5 6 7 8
YOUR MOVE FROM(LINE, COLUMN) ? 4,3
TO ? 5,2

I CAN'T MOVE
X O X O X O X O  L1
H X O X O X O X  L2
X H X O X O X O  L3
F X O X O X O X  L4
X H X H X O X O  L5
O X O X O X O X  L6
X O X O X O X O  L7
O X O X O X O X  L8
1 2 3 4 5 6 7 8
YOU WIN...WANT TO PLAY AGAIN ? NO

READY
```

Sample run.

See Publisher's Remarks, No. 18, re Baudot.

Baudot Interface Cookbook

A $50 printer? A simple junk-box interface and 500 bytes can put any Baudot teleprinter on line.

The Teletype model 15 printer and stand, harnessed to an 8080.

J. R. Haglund
W. B. Reed
NADS Engineering
4401 SE Rockwood
Milwaukie OR 97222

After spending nearly two thousand dollars on our microcomputer system, we still didn't have a hard-copy device. Since we spend a lot of time writing, debugging and running programs (mostly debugging), we decided that we definitely needed a printer.

Cost was the major criterion in our search for the ideal printer, but reliability, flexibility and at least 64-character-per-line capability were also considerations. Thumbing through several small-computer magazines, we found that the available printers ranged in price from $250 for a 40-character dot-matrix printer (unacceptable) to $900 for a reworked ASR Teletype* (too expensive) to over $1500 for a fancy line printer (dream on). There had to be a cheaper way to get hard copy.

An extensive computer analysis of the cost/ performance trade-offs yielded a system that has cost and simplicity in its favor. The system generates

---

*Teletype—Trademark, Teletype Corp.

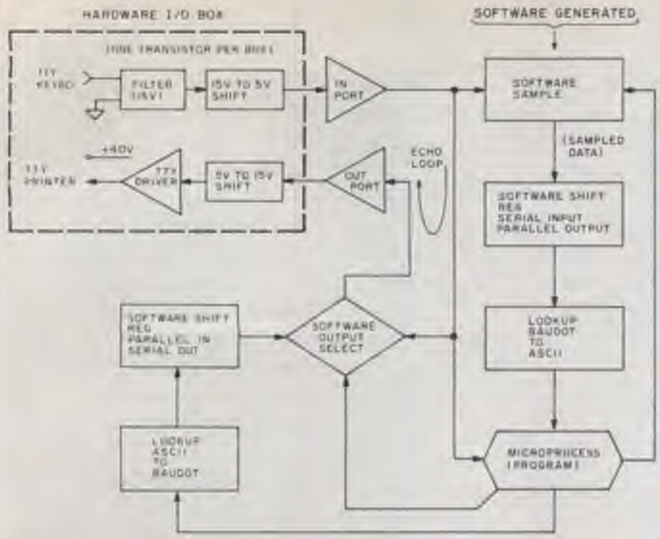


Fig. 1. General view of how the system works. The only hardware is inside the dash-outlined box. Note that the echo loop bypasses the main program to allow maximum Baudot keyboard input speed.

the hard copy utilizing a Baudot teleprinter—the standard machine of the news and wire-service (they're often heard banging away during local news broadcasts). They are available on the surplus market in prices ranging down from $50.

System Description

We have developed an inexpensive interface that consists of one Baudot teleprinter, a hardware input/output box containing only a few resistors and transistors and a software package consisting of about 500 bytes in 8080 code. The required low cost is achieved by use of virtually free timed software to receive, decode and transmit the serial Baudot data. This reduces the need for expensive parallel-to-serial hardware logic. Refer to Fig. 1 for a general flowchart of the system.

Features of the interface are:

1. Low cost—under $20 for parts and wire if you must purchase *everything*.

2. Simple hardware—uses a basic two-transistor circuit for the required input and output drivers.

3. Simple software—consists of 500 bytes of 8080 code and may be adapted to system requirements.

4. Minimal modifications—software eliminates expensive parallel-to-serial converters and utilizes a spare bit on existing input and output ports.

5. Expanded character set—software simulates output of most required math characters not existing on the Baudot, without any hardware modifications.

6. Baudot keyboard input—during program executions or input of data, the interface conveniently accepts Baudot keyboard data.

7. STOP—program execution may be stopped by depressing the Baudot *break* key.

8. Punched tape—several Baudot machines have tape drives and punches, and there are separate perforation/readers available.

Sound too good to be true? It is! This is the cheapest hard copy you can find; it requires almost no hardware and outputs an expanded character set. We will touch upon the basics of the Baudot machine and present all detailed documentation required for the implementation of the system. Also included are certain start-up and troubleshooting hints.

How the Baudot Printer Works

The Baudot teleprinter is a serial device; each letter is a set of five binary-code bits (not including the start and stop bits) as shown in Fig. 2. (The letter R is shown for reference.) Note that start bits are always logic low; stop bits are always logic high and stay that way when no data is present.

Baudot machines have a limited character set (see Fig. 3) and are not compatible with ASCII. Since math symbols are a must with most programs, we used some software tricks in the interface. Fig. 4 shows the expanded ASCII/Baudot character set chosen for our particular requirements. Be forewarned that there are some minor differences between uppercase Baudot characters. On our three machines there are three figures, the pound sign, apostrophe and STOP, that are interchanged. Simple modifications of the software lookup table, described later, will solve this problem.

One beautiful feature found on the Teletype model 19 is the STOP button. When the STOP code is sent, the machine turns itself off, and upon receiving a low bit, the machine will automatically power-on! This allows for total remote control from the computer terminal.

A quick review of the workings inside the machine is now in order. The serial binary pulses open and close a set of selector magnets, which in turn select the proper character to by typed. Current flowing through the magnets holds the selector bars from moving; the absence of current allows them to begin selecting a character. In effect, each code bit opens and closes the magnets, mechanically selecting a character. All code bits are the same time length regardless of how fast the printer is operating. Faster typing simply generates more of the five bit words per given time period, up to 60 wpm. The keyboard has a set of five switches that run off five timing cams. When depressed, each letter selects the proper array of cams to open and close the switches, generating the code word.

Unfortunately, the Baudot machine has several limitations compared to ASCII. First, numerals and special characters are all in the uppercase mode. Typing a period, for example, takes three discrete moves. Fortunately, the computer does all the shifting when ASCII keyboard or output is run through the program. Second, no math characters exist on the Baudot. As mentioned, Fig. 4 shows our equivalent set.

Third, most Baudots run at 60 wpm (45 baud), at which some hard-core computer freaks will scoff. Since most of us have more time than money, 60 wpm is perfectly acceptable. Beware when selecting a machine—several different speeds are available. Sixty wpm printers comprise the vast majority, but some 75 and 100 wpm machines may be found. The output program will work on all; however, one must change the software timing cycles to sample the faster data.

A word to the wise: If you are starting this from scratch, you had better buy a reference book on teleprinters. (Books may be found in most ham radio stores.) A few special adjustments and problems that simply cannot be covered in the scope of this article may exist in the printer.

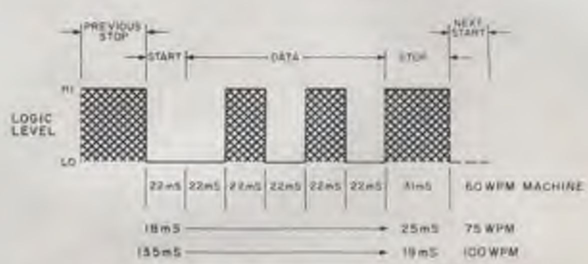


Fig. 2. The Baudot timing diagram consisting of five binary-code words and the start and stop pulses. Shaded portions represent current flow through the printer magnets. Times for higher-speed machines are shown for reference.

Software

The software accepts, processes and transmits the serial Baudot data. The three main programs, Status, Input and Output, are designed so that they may be easily interfaced to the input and output schemes of your BASIC interpreter or assembler, which will be described later. The functions of the Status, Input and Output routines are described below. It should be noted that the software allows input from both an ASCII keyboard and the Baudot teleprinter. Also, the program will output to either or both the printer and a video monitor.

Status. Status checks for data availability on both the Baudot printer and the ASCII keyboard as shown on the flowchart in Fig. 5. If data is available, Status returns with the zero condition bit reset; if data is not available, the zero condition bit is set. This convention is compatible with most software and hardware configurations used today. Status also keeps track of at which keyboard the data availability occurred; this information is used by the Input routine.

Data availability occurs on the ASCII keyboard when the input strobe from the keyboard forces a bit to go low on the input status port. (Note: If your status bit normally forces the status bit to a high logic level, the CMA operation shown at address F007 should be changed to a NOP.)

Data availability occurs on the Baudot when the normally active stop signal (logic high) changes to a start signal (logic low) for a duration of 22 ms. Status, therefore, looks for the input signal to go low. Since an unknown interference could have momentarily caused a low signal, the routine samples the signal 162 times during the 22 ms start interval. If at least half of these samples are low, the routine returns with data-availability status. If not, the routine decides that the start signal was not legitimate and returns with the data-not-available status.

Input. The function of the Input routine is to return with the ASCII-encoded data in the accumulator register (see the flowchart in Fig. 6). Input first checks which device—either the Baudot or the ASCII keyboard—generated the data-available signal using a flag provided in the Status routine.

If the ASCII keyboard is guilty, the procedure is trivial. The routine simply inputs off the keyboard port, masks off the parity (most significant) bit and returns with this in the accumulator.

If the source is the Baudot machine, the procedure is a little more complicated. The first step is to accept the incoming Baudot data. Each of the five bits that define the Baudot code are of the same 22 ms duration. The timed software samples the logic state of each bit 163 times during the 22 ms interval. Since there may have been extraneous noise (a remote possibility) during transmission, the true logic state is decided by a democratic process called plurality or majority rule. As each bit is received it is put sequentially into a software shift register until all five data bits are received. Additionally, during sampling, the input signal is echoed directly back to the teleprinter to allow for instant keyboard-printer operation.

Each key on the Baudot has an alphabetic and figure mode driven by keyboard shifts prefixed by the LET and FIG keys respectively. Therefore, the software must check the previous state of the carriage mode before it can begin to determine the ASCII equivalent.

Once the mode is established, the software uses the lookup table (described later) to

	Baudot Character		Baudot code Signal	
	Lowercase	Uppercase	Binary	Hex
	A	-	00011	03
	B	?	11001	19
	C	:	01110	0E
	D	$	01001	09
	E	3	00001	01
	F	!	01101	0D
	G	&	11010	1A
	H	#	10100	14
	I	8	00110	06
	J	'	01011	0B
	K	(	01111	0F
	L	)	10010	12
	M	.	11100	1C
	N	,	01100	0C
	O	9	11000	18
	P	0	10110	16
	Q	1	10111	17
	R	4	01010	0A
	S	BELL	00101	05
	T	5	10000	10
	U	7	00111	07
	V	;	11110	1E
	W	2	10011	13
	X	/	11101	10
	Y	6	10101	15
	Z	"	10001	11
	Space		00100	04
	Carriage Return		01000	02
	Line Feed		00010	02
Shift	Uppercase		11011	1B
	Lowercase		11111	1F

Fig. 3. Baudot character set as found on the machine. Some differences may exist in the uppercase characters. In this scheme, least significant bits in the binary table are to the right.

INPUT	OUTPUT	ASCII ⇌	BAUDOT
✓	✓	+	&
	✓	>	GT
	✓	<	LT
	✓	=	EQ
✓	✓	*	X
✓		CNTRLC	BREAK (in letters mode)
✓		DELETE	BREAK (in numbers mode)

Fig. 4. The expanded character set chosen for the interface. Note that not all characters may be used both on input and output.

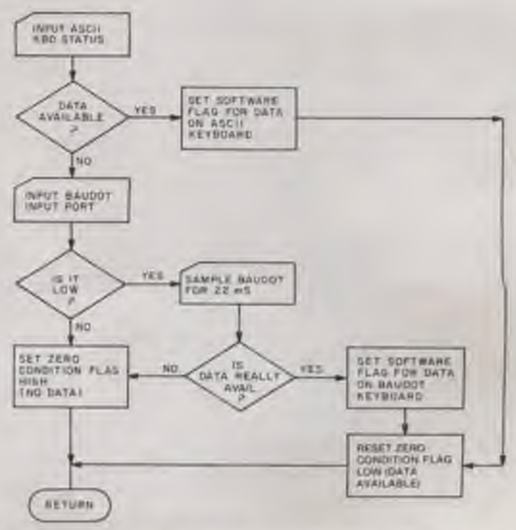


Fig. 5. Status routine flowchart establishes which keyboard has input data.

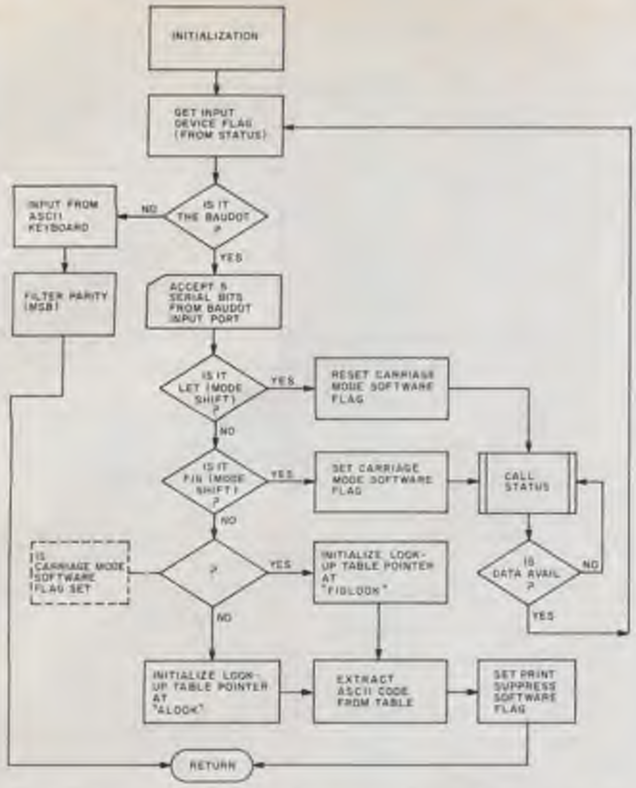


Fig. 6. Input routine flowchart represents logic followed by computer in setting up the Baudot input, finding letters or characters and returning ASCII to main programs.

extract the ASCII value of the Baudot code. Note that the input from the Baudot is limited by the machine's hardware (exceptions are noted in Fig. 4). For example, lowercase letters are not possible. Input from the Baudot is mainly a convenience feature for inputting data during program executions and Star Trek marathons. It is assumed that the ASCII keyboard will be the one predominantly used due to its simplicity of operation.

To simulate mathematical symbols not found on the Baudot keyboard, certain Baudot characters were sacrificed. The plus and * operations were converted to & and X, respectively, during all interactions with the Baudot machine. As an added feature, the BREAK (NOP) key found on most machines is used to generate an ASCII CONTROL C in the letter mode or an ASCII DELETE when in the figure mode.

Output. The output routine converts the ASCII data in the accumulator to Baudot code and outputs it serially to the Baudot machine (see Fig. 7). The Baudot code is obtained via the lookup table (explained in detail in the lookup table section). Since each Baudot code represents two possible characters depending on the status of the carriage shift, the software checks and, if necessary, changes the carriage shift by generating the LET or FIG code as appropriate. For example, if the teleprinter is in letters mode and the computer wants to type a 5, the software must generate a shift character (the FIG key) prior to generating the Baudot code for a 5.

Before outputting the five-bit Baudot code, the software sets up the hardware-required start and stop signals immediately preceding and following the code. This is done by putting the binary sequence 11DDDDD0 in a register. D represents the Baudot code; 0 is the start signal (logic low) and 11 stands for the stop signal (logic high). This register is then shifted out one bit at a time 22 ms per bit to the output port. At this point, the I/O box takes over and drives the magnets of the Baudot machine to generate the required character.

Since the teleprinter does not have >, < and = in its character set, the software converts these to GT, LT and EQ (pseudo-FORTRAN fashion).

Lookup table. To make it extremely simple and efficient, the lookup table is divided into two sections starting at addresses ALOOK and FIGLOOK. Using the Baudot code as the base during input, the code word's numeric value is added to the address of ALOOK or FIGLOOK depending upon the status of the carriage shift. The contents of this address is the ASCII code. To clarify, assume that the Baudot carriage is shifted to figures mode and you type an 8 on the Baudot. The Baudot code for 8 (see Fig. 3) is binary code 00110 or 6. The contents of FIGLOOK plus 6 is 38 hex, which corresponds to the ASCII code for 8.

To determine the Baudot code for a given ASCII code (during output), the opposite procedure is used. Starting at address ALOOK, a search is made for the ASCII code. When a match occurs the address bias is removed, and the Baudot code remains.

Sampling technique. Earlier versions of this software used a single sample at the center of each 22 ms data bit transfer during input from the Baudot machine. Although this method was nearly 100 percent reliable for our system, we realized that for general systems it was probably too susceptible to glitches. We developed a more refined technique using multiple sampling.

During each 22 ms bit transmission period, the logic state on the input bit is sampled 163 times, and the state of the data is accumulated in a counter until the interval is over. If the count is greater than 81, it is concluded that the bit is a logic 1. If not, the bit is assumed to be a logic 0. This software feature, in turn with the hardware design features, effectively

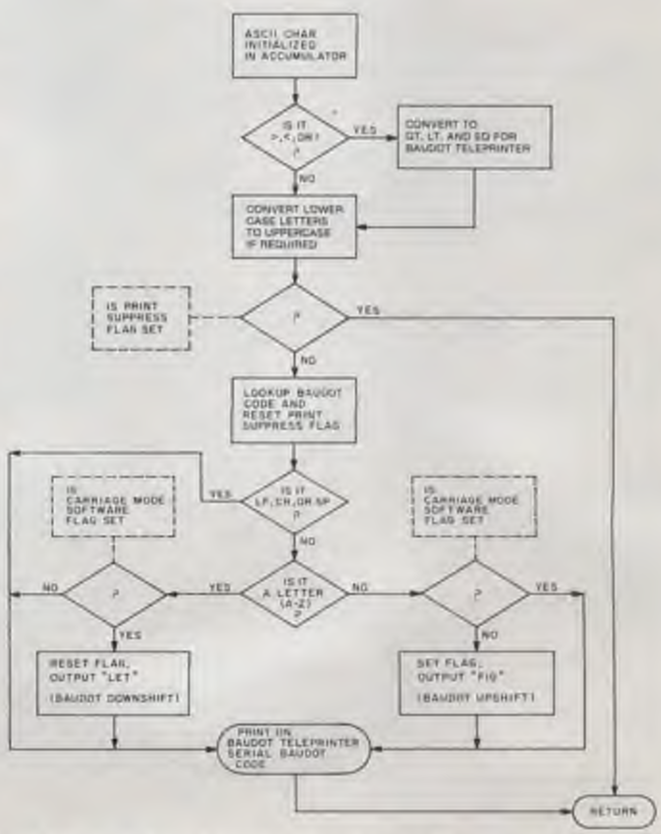
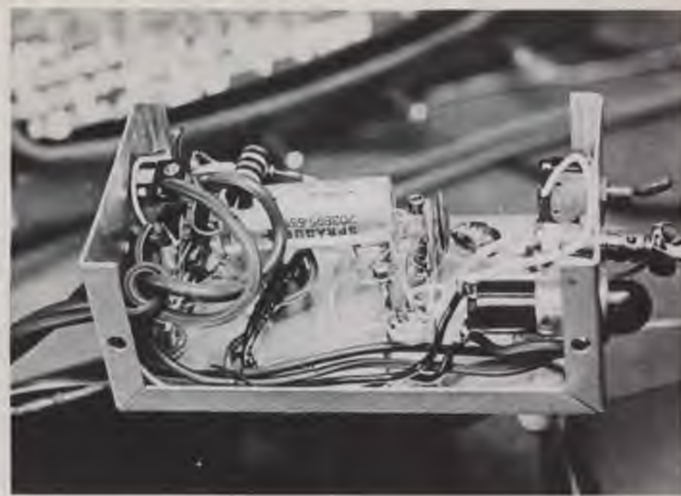


Fig. 7. Output flowchart shows the ASCII to Baudot conversion, generation of special characters and software serial shift out.



Close-up of the hardware box required for I/O. This prototype box was assembled on a recycled chassis from junk-box parts. Note that the heavy cable to the printer enters on the left near the output driver, while the input driver and its cables are on the right.

mitigates the effect of any stray signals.

At the same time the input data is sampled, it is echoed back to the teleprinter. This allows the typist to go at full 60 wpm speed since the echo is part of the input routine. The Output software suppresses echo during output in order to prevent double echo. We decided that this scheme was more acceptable than having the software process the serial input and then echo it back in the Output routine. This method would effectively cut speed to 30 wpm and result in annoying (as found from experience) printer delays.

Software/hardware compatibility. Since hardware I/O configurations differ widely, the software has provisions to allow it to be customized to individual systems. Your configuration will probably not be the same as ours; therefore, look at the following addresses to make necessary modifications.

ASCII keyboard:

1. Address F006 contains the ASCII keyboard status port.

2. Address F009 contains the ASCII keyboard status bit.

3. Address F007 should be changed to 00 if your status bit goes high on data availability.

Baudot teleprinter input:

1. The input signal from the I/O box is put into a spare bit on a parallel port. Note: A synchronous logic gate bit (e.g., an 8212) won't work. You must use a status type bit that can follow asynchronous logic. We used a spare status bit on our Tarbell cassette interface. The software requires that the incoming signal be rotated to bit 0 (LSB) in the Inshift subroutine (address F046). Plenty of NOPs have been provided for this purpose. Naturally, these must be changed to ROTATE instructions as necessary. A CMA instruction may also be required by your particular system at address F04C.

2. Address F047 should be changed to the address of your Baudot input port.

Baudot teleprinter output:

1. Output to the teleprinter occurs in a subroutine called Outshift. The signal to be transmitted enters the subroutine in bit 0 (LSB). If the teleprinter output bit differs from this, you must add the necessary rotates to set the data in the proper bit location.

2. Our output hardware is such that the logic state on the output bit was inverted. This required a CMA at location F052. This CMA may be changed to a NOP for your hardware.

3. Address F054 should be changed to the address of your Baudot output port.

Computer speed and wait states:

This software* was written in 8080 code for use with a system utilizing a 2 MHz frequency (500 ns cycle time) and no wait states. If your system differs, certain critical timing parameters must be changed to accommodate your system. Fig. 8 shows what these timing parameters should be for a variety of possibilities. Incidentally, this software is PROMable with the exception of the program flags at locations PORTSEL, MODFL and SUPFL.

Video driver:

A patch is provided at address F16E for calling a video driver or another output driver (another Baudot?). The contents of addresses F16F and F170 should be changed to the location of your driver to prevent an immediate bomb.

Hardware I/O Box

As intended, the hardware I/O box is a trivial driver circuit (see Fig. 9). First, a giant shortcut was taken with the driver: Only 40 V dc is used to run the printer magnets. Traditionally, 150 V dc is used in the older teleprinter circuits. 40 V dc is the absolute minimum and works, basically because the pulse timing and noise levels of the computer are excellent compared to what is required to run the machine. This lower voltage allows for cheaper and safer operation of the machine. Both a model 15 and 19 Teletype and a Western Union teleprinter were tested and found to work perfectly off this voltage.

Moving on—the output portion of the circuit is a simple Darlington amplifier and level shifter. Although such a high-gain circuit is not required with high-quality exacting-specification transistors, such components can rarely be found in the average junk box. The diode across the output collector circuit protects the transistor from inductive kickback voltages produced by the printer magnets.

Next, the input circuit is also an amplifier-level shifter. The capacitor and resistor input circuit is a tuned PI network, allowing passage of Baudot characters while preventing feedback of stray printer glitches, computer radiation and CB "good buddies" from bombing the CPU. Don't vary the timing circuit components too much or you may also remove the Baudot input!

**Program has been tried on a Z-80 CPU and works without any modifications.*

Your system			Change these addresses as shown						
Teleprinter Speed	Computer Cycle Time	Wait States	F017	F01C F06E	F069	F0B2 F168	F0B3 F169	F18E	F18D
60 wpm	0.5 us	0	A2	51	A3	44	48	59	06
60	0.5	1	82	40	81	6C	39	09	05
60	0.5	2	6B	35	6A	9E	2F	2C	04
60	1.0	0	52	29	51	22	24	29	03
60	1.0	1	41	20	40	B6	1C	82	02
75 wpm	0.5 us	0	83	41	82	DA	39	12	05
75	0.5	1	68	34	67	F0	2D	06	04
75	1.0	0	42	21	41	D8	1C	86	02
100 wpm	5.0 us	0	63	31	62	5C	2B	CC	03
100	0.5	1	4E	27	4D	74	22	02	03

Fig. 8. Examples of timing changes required for different computer cycle times.

OSBORNE & ASSOCIATES, INC. O6

The World Leaders In Microprocessor Books

If you want information on microprocessors, begin with the Osborne books.

ASSEMBLY LANGUAGE PROGRAMMING

8080A/8085 Assembly Language Programming

6800 Assembly Language Programming

These books describe how to program a microcomputer using assembly language. They discuss classical programming techniques, and contain simplified programming examples relevant to today's microcomputer applications.
#31003, 32003 (400 pages each)

PROGRAM BOOKS WRITTEN IN BASIC

Payroll With Cost Accounting
Accounts Payable And Accounts Receivable
General Ledger

These books may be used independently, or implemented together as a complete accounting system. Each contains program listings, user's manual and thorough documentation. Written in an extended version of BASIC.
#22002 (400 pages), #23002 , #24002*

Some Common BASIC Programs

76 short practical programs, most of which can be used on any microcomputer with any version of BASIC. Complete with program descriptions, listings, remarks and examples.
#21002 (200 pages)

AN INTRODUCTION TO MICROCOMPUTERS

Volume 0 - The Beginner's Book

If you know nothing about computers, then this is the book for you. It introduces computer logic and terminology in language a beginner can understand. Computer software, hardware and component parts are described, and simple explanations are given for how they work. Text is supplemented with creative illustrations and numerous photographs. Volume 0 prepares the novice for Volume I. #6001 (300 pages)

Volume I — Basic Concepts

This best selling text describes hardware and programming concepts common to all microprocessors. These concepts are explained clearly and thoroughly, beginning at an elementary level. Worldwide, Volume I has a greater yearly sales volume than any other computer text. #2001 (350 pages)

Volume II — Some Real Products (revised June 1977)

Every common microprocessor and all support devices are described. Only data sheets are copied from manufacturers. Major chip slice products are also discussed.
#3001A (1250 pages)

PROGRAMMING FOR LOGIC DESIGN

8080 Programming For Logic Design
6800 Programming For Logic Design
Z80 Programming For Logic Design

These books describe the meeting ground of programmers and logic designers; written for both, they provide detailed examples to illustrate effective usage of microprocessors in traditional digital applications.
#4001, #5001, #7001 (300 pages each)

	These prices effective July 1, 1978.	PRICE	QTY	AMT
6001	Volume 0 — The Beginner's Book	$ 7.95		
2001	Volume I — Basic Concepts	$ 8.50		
3001A	Volume II — Some Real Products	$15.00		
4001	8080 Programming for Logic Design	$ 8.50		
5001	6800 Programming for Logic Design	$ 8.50		
7001	Z80 Programming for Logic Design	$ 8.50		
31003	8080A/8085 Assembly Language Programming	$ 8.50		
32003	6800 Assembly Language Programming	$ 8.50		
21002	Some Common BASIC Programs	$ 8.50		
22002	Payroll With Cost Accounting	$15.00		
23002	Accounts Payable and Accounts Receivable	$15.00		
			TOTAL	
			Sales Tax (Calif. residents only)	
			Shipping Charges	
			TOTAL AMOUNT ENCLOSED	

- 6½%, SF Bay Area residents only
- 6%, California residents outside SF Bay Area
- **Payment by check or money order must be enclosed for orders of 10 books or less.**

OSBORNE & ASSOCIATES, INC.
P.O. Box 2036 DEPT. D5
Berkeley, California 94702
(415) 548-2805
TWX 910-366-7277

NAME

ADDRESS

CITY

STATE ZIP PHONE

SHIPPING CHARGES: Shipping for large orders to be arranged.

- ☐ All foreign orders, $3.00 per book, for air shipment
- ☐ 4th class $0.35 per book (allow 3-4 weeks within USA, not applicable to discounted orders)
- ☐ $0.75 per book, UPS (allow 10 days) in the U.S.
- ☐ $1.50 per book, special rush shipment by air in the U.S.

Please send information on:

- ☐ Becoming an O&A dealer
- ☐ School discounts
- ☐ List of foreign distributors

***This book is scheduled to be published during 1978**
Please notify me when it is available:

- ☐ 24002 General Ledger

F8

Finally, the requirements of the computer in the I/O schemes are trivial. Only one spare bit on an input and output port is required since the computer generates the serial information. Depending on the nature of your particular output port, a pull-up resistor may be required on the I/O card itself, as shown in the schematic.

That's about it for the I/O box. As we mentioned before, the hardware was kept to a minimum as this tends to be the major stumbling block when people home-build equipment.

Debug and Construction

So now you're geared up to build the interface. Refer to the main schematic diagram (Fig. 9). We recommend that a separate box be used for the I/O hardware to ward off interference between the teleprinter and computer. Anytime one attempts to interface two systems that were designed over 120 years apart, certain incompatibilities are bound to exist (the first teleprinter came on line in 1849!).

Component placement is not critical in the box, and the quickest placement method is to use tie-points or vector board construction. In our opinion, a printed circuit board for such a simple circuit is a waste of time. A good construction practice is to keep the input and output sections physically separated. Lead in and out wires should also enter as far apart as possible. Shielded wire runs between the box, computer and machine would be a good idea but are only required on the computer input as shown. Keep the box-to-computer input port cable as short as possible (two feet maximum); it must be shielded.

The transistors may be any junk-box variety having at least three good leads. To be on the safe side, we used a 1 Amp power transistor for the output. While a small-signal-type transistor will drive the magnets, it would not survive any accidental short circuiting. All other transistors are general-purpose NPN silicon. The only critical parts are those in the input filter.

Components

RFC. This is not critical, just about any rf choke in the 1 to 10 millihenry range will do.

R1. This may be a difficult component to calculate, since the voltage-current requirements may vary with different teleprinters. With the Teletype model 15 and 19, the resistor is 270 Ohms, 2 Watts. If you are not sure what your machine will need, use a 1000 Ohm, 5 Watt adjustable wire-wound resistor and set it to the proper current (20 to 60 mA on most printers).

Q1. Use a 1 Amp power NPN silicon transistor with a maximum collector voltage of 40 V or greater as required by transformer voltage.

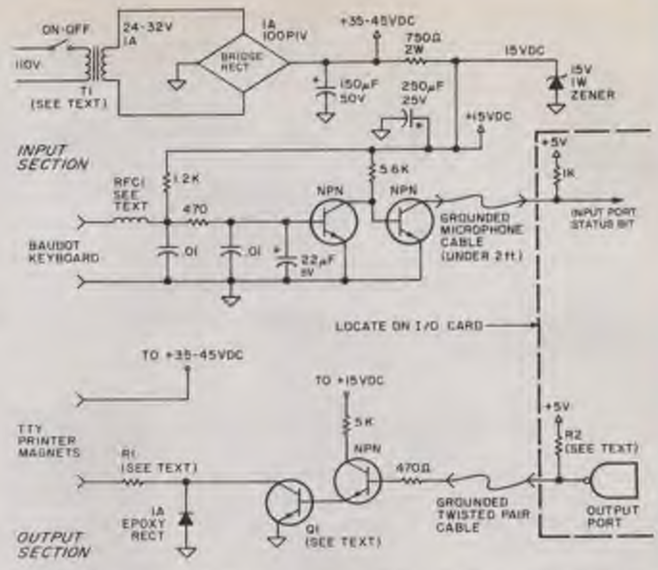


Fig. 9. The trivial hardware required for this interface. Components, with the exception of those in the Baudot keyboard input section, are not critical. Use ¼ Watt resistors except as noted. Also refer to the photo of the interface box for an example of layout.

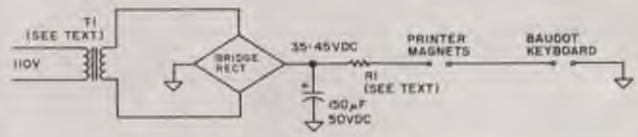


Fig. 10. A simple test circuit for initial testing and cleanup of a "new" junk Baudot printer.

R2. Use a 3000 Ohm, ¼ Watt resistor if your port is an open-collector-type output. If you're not sure, use a voltmeter and test for an output voltage at logic high, indicating that it is not the open type.

T1. You need a nominal 24 V, 1 Amp filament transformer. The Radio Shack 25.2 V transformer is an ideal less expensive model. If you don't have a transformer in the junk box and wish to order one, try to buy a 26 to 32 V transformer.

Start-up and Testing.

If this is your first experience with a teleprinter, set up the test circuit shown in Fig. 10 and test for proper operation of all characters. Typing RYRYRY is a good test for proper internal timing. In the rare case that your printer refuses to operate on the 40 V circuit (due to old age or stiff joints), either get a new junk machine or look at higher voltage driver circuits from a book on traditional circuits. (Also, check the *73 Magazine Teleprinter Handbook* for ideas on advanced low-voltage driver circuits.)

When satisfied that the printer will talk to itself properly, move on to Test Program 1. This program tests the hardware by simply accepting the serial data on the input port and echoing it back to the printer. Load

```
0000                    0010 X
0000                    0020 X                        TEST PROGRAM 1
0000                    0030 X
0000                    0040 X  THIS PROGRAM TESTS THE
0000                    0050 X  INPUT / OUTPUT BOX AND
0000                    0060 X  THE I/O PORTS FOR PROPER
0000                    0070 X  OPERATION USING A SIMPLE
0000                    0080 X  ECHO ROUTINE
0000                    0090 X
0000  DB  6E            0100 LOOP IN IPORT            INPUT PORT
0002  00                0110     NOP
0003  00                0120     NOP                  FOR ROTATES IF REQ'D
0004  00                0130     NOP
0005  00                0140     NOP
0006  2F                0150     CMA                  FOR CMA IF REQ'D
0007  D3  63            0160     OUT OPORT            OUTPUT PORT
0009  C3  00  00        0170     JMP LOOP
000C                    0180 IPORT EQU 6EH            TARBELL SPARE
000C                    0190 OPORT EQU 63H            TARBELL SPARE
000C                    0200 X
000C                    0210 X
```

Fig. 11. Test Program 1.

the program shown in Fig. 11 and type from the Baudot keyboard. This program will also test for logic continuity. If your I/O scheme inverts the logic, the printer will jump all over the floor. Simply invert the logic through software in both the test and main programs. Provisions are made for this modification in the subroutines labeled Inshift and Outshift.

When the first program runs satisfactorily, load in the main programs and make any compatibility adjustments as necessary. Next, load and execute Test Program 2 (Fig. 12), which will allow input from either the ASCII or the Baudot keyboard and echo to the video monitor and printer. Test for problem-free input echo both on the screen and teleprinter. Any problems will point to timing-loop errors or minor printer-timing errors. Some printers have fine-timing adjustments easily tuned to small timing errors. Consistently incorrect characters most likely would be a result of lookup table differences between your teleprinter and ours.

When these tests are complete, you are home free and may proceed to modify your software packages to output hard copy.

Software Interface

If you've made it this far, you are ready to interface the Status, Input and Output routines to your BASIC interpreter or assembler. The first step is to locate the input and output driver sections of your software and modify them as required. Sample modifications are shown in Fig. 13.

Conclusion

Although we've given you the basics of a good, inexpensive hard-copy device, there are many possibilities for future expansion. One such application would be total remote-access capabilities using a more comprehensive character set. Since amateur-radio operators have been using Baudot machines for years, perhaps one may wish to put a computer on the airwaves for Baudot time share. the same principal may be used over telephone lines.

Although this interface was designed for the Baudot machine, modifications to the lookup table and timing cycles would allow it to be used with any device requiring binary serial data.

If you have questions or problems with our interface, feel free to contact us. We'll provide tapes of the object code assembled to your addressing requirements using the Tarbell or Kansas City Standard (CUTS) interface. Send a $5 check ($8 for North Star minifloppy disk) payable to NADS Engineering.■

A Consumers' Guide to Baudots

As a quick guide on which machines to look for, we present the following older printers. Check ham-radio-store bulletin boards for these models. $50 or less would be a good price for the older machines.

Teletype model 15. The standard of the news services and chosen for our interface.

Teletype model 19. Similar to the model 15, except with auto start/stop and tape punch.

Teletype models 28 ASR & KSR. These units have capabilities of 75 and 100 wpm operation. As these units are newer, a higher price may be paid.

Western Union model 104. We don't have the rundown on Western Union's printers, but this one was given to us and is an excellent, light and quiet printer.

Teletype model 26. Though this unit was not tested with our interface, it should work. These are small and lightweight, with a type cylinder similar to modern printers.

```
0000                  0010  X
0000                  0020  X                      TEST PROGRAM 2
0000                  0030  X
0000                  0040  X  THIS PROGRAM TESTS THE THREE
0000                  0050  X  PROGRAMS (STATUS, INPUT, AND
0000                  0060  X  OUTPUT) FOR PROBLEMS ASSOCIATED
0000                  0070  X  WITH HARDWARE COMPATIBILITY
0000                  0080  X  AND TIMING.
0000                  0090  X
0000  31   00   01    0100     LXI SP,100H
0003  CD   00   F0    0110  LOOP CALL STATUS
0006  CA   03   00    0120     JZ LOOP
0009  CD   56   F0    0130     CALL INPUT
000C  CD   C4   F0    0140     CALL OUTPUT
000F  C3   03   00    0150     JMP LOOP
0012                  0160  SP EQU 6
0012                  0170  STATUS EQU 0F000H
0012                  0180  INPUT EQU 0F056H
0012                  0190  OUTPUT EQU 0F0C4H
0012                  0200  X
0012                  0210  X
```

Fig. 12. Test Program 2.

TYPICAL			CHANGE TO	
A. STATUS	LOOP	IN 0	LOOP	CALL STATUS
		ANI 40H		JZ LOOP
		JZ LOOP		NOP
B. INPUT		IN 01		CALL INPUT
		ANI 7F		NOP
C. OUTPUT TO PRINTER	LOOP	IN 0		CALL OUTPUT
		ANI 80H		NOP
		JZ LOOP		NOP
		OUT PORT		NOP
				NOP
				NOP
D. OUTPUT TO VIDEO MONITOR	LOOP	CALL VDM		CALL OUTPUT

Fig. 13. Examples of modifications required to interface software with existing system programs.

Baudot to ASCII interface program listing.

```
F000                  0005  X
F000                  0010  X
F000                  0013  X ---------------------------------
```

```
F000                0016 X
F000                0018 X        BAUDOT TO ASCII INTERFACE
F000                0020 X
F000                0022 X        DESIGNED AND DEVELOPED BY
F000                0023 X              NADS ENGINEERING
F000                0024 X
F000                0025 X     SOFTWARE BY JOHN R HAGLUND
F000                0027 X     HARDWARE BY WILLIAM B REED
F000                0028 X
F000                0030 X              AUGUST 1, 1977
F000                0032 X
F000                0034 X  --------------------------------
F000                0036 X
F000                0037 X  STATUS INPUT ROUTINE
F000                0040 X         - CHECKS FOR DATA AVAILABILITY
F000                0042 X           ON EITHER AN ASCII KEYBOARD
F000                0044 X           OR A BAUDOT TELEPRINTER
F000                0046 X
F000 D5             0050 STATUS PUSH D
F001 E5             0060    PUSH H
F002 C5             0070    PUSH B
F003 1E  01         0080    MVI E,1
F005 DB  00         0090    IN KBDSTATUS         ASCII KBD STATUS
F007 2F             0100    CMA                  ONLY IF BIT LOW
F008 E6  01         0120    ANI BIT
F00A C2  24  F0     0130    JNZ DAV              DATA ON KBD
F00D CD  46  F0     0140    CALL INSHIFT         INPUT ON TTY PORT
F010 2F             0150    CMA
F011 E6  01         0160    ANI 1
F013 CA  22  F0     0170    JZ NODATA
F016 16  A2         0180    MVI D,162            22 MS SAMPLE
F018 CD  2C  F0     0190    CALL SAMPLE
F01B 3E  51         0200    MVI A,81             IF L LT 22 THEN DATA
F01D 95             0210    SUB L
F01E 1F             0212    RAR
F01F 2F             0215    CMA
F020 E6  80         0220    ANI 80H
F022 1E  02         0270 NODATA MVI E,2          IMPLIES TTY
F024 21  D8  F1     0280 DAV LXI H,PORTSEL
F027 73             0290    MOV M,E              SELECT INPUT DEVICE
F028 C1             0310    POP B
F029 E1             0320    POP H
F02A D1             0330    POP D
F02B C9             0340    RET
F02C                0350 X
F02C                0360 X  SAMPLE ROUTINE
F02C                0362 X         - SAMPLES AND ECHOS TO THE
F02C                0364 X           BAUDOT MACHINE THE INCOMING
F02C                0366 X           SERIAL CODE. L IS COUNTER
F02C                0369 X
F02C 2E  00         0400 SAMPLE MVI L,0          COUNTER
F02E CD  46  F0     0420 LOOP CALL INSHIFT
F031 E6  01         0430    ANI 1
F033 47             0440    MOV B,A
F034 85             0450    ADD L
F035 6F             0460    MOV L,A
F036 78             0465    MOV A,B
F037 CD  4E  F0     0470    CALL OUTSHIFT
F03A CD  46  F0     0474    CALL INSHIFT
F03D CD  4E  F0     0476    CALL OUTSHIFT
F040 15             0480    DCR D
F041 C2  2E  F0     0510    JNZ LOOP
F044 42             0515    MOV B,D
F045 C9             0520    RET
F046 DB  6E         0600 INSHIFT IN IPORT
F048 00             0610    NOP
F049 00             0620    NOP
F04A 00             0630    NOP
F04B 00             0640    NOP                  ROTATE TO BIT 0
F04C 00             0650    NOP                  FOR CMA IF REQ'D
F04D C9             0660    RET
F04E 00             0700 OUTSHIFT NOP
F04F 00             0720    NOP
F050 00             0730    NOP
F051 00             0740    NOP                  ROTATE TO OUTPUT BIT
F052 2F             0750    CMA                  FOR CMA IF REQ'D
F053 D3  6E         0760    OUT OUTPORT
F055 C9             0770    RET
F056                0800 X
F056                0810 X  ------------------------------------
F056                0820 X
F056                0830 X  INPUT ROUTINE
F056                0840 X         - ACCEPTS INPUT FROM AN ASCII
F056                0850 X           ENCODED KEYBOARD OR . . .
F056                0860 X         - CONVERTS INCOMING SERIAL BAUDOT
F056                0870 X           CODE TO ITS ASCII EQUIVALENT
F056                0880 X
F056                0890 X  ------------------------------------
F056                0900 X
F056 D5             1000 INPUT PUSH D
F057 E5             1020    PUSH H
F058 C5             1030    PUSH B
F059 21  D8  F1     1040 PS LXI H,PORTSEL
F05C DB  01         1050    IN KBD               ASCII KEYBOARD
F05E E6  7F         1060    ANI 7FH              CLEAR PARITY
F060 35             1070    DCR M
F061 CA  C0  F0     1080    JZ INRET             DATA ON ASCII KEYBD
F064 21  00  05     1090    LXI H,500H           H IS COUNTER
F067 E5             1100 ILOOP PUSH H            L IS SHIFT REG.
F068 16  A3         1110    MVI D,163            22 MS COUNTER
F06A CD  2C  F0     1120    CALL SAMPLE
F06D 3E  51         1130    MVI A 81             IF L GT 97 THEN DATA HI
F06F BD             1140    CMP L
F070 1F             1150    RAR                  PUT CARRY IN MSB
F071 E6  80         1160    ANI 80H
F073 E1             1170    POP H
F074 B5             1180    ORA L
F075 OF             1190    RRC
F076 6F             1210    MOV L,A
F077 CD  46  F0     1214    CALL INSHIFT
F07A CD  4E  F0     1216    CALL OUTSHIFT
F07D 25             1220    DCR H
F07E C2  67  F0     1230    JNZ ILOOP
F081 7D             1240    MOV A,L
F082 0F             1250    RRC                  RIGHT JUSTIFY
F083 0F             1260    RRC
F084 4F             1270    MOV C,A              SAVE IN C
F085 FE  1B         1280    CPI 1BH              FIG CHAR?
F087 CA  8F  F0     1290    JZ MODESW
F08A FE  1F         1310    CPI 1FH              LETTER CHAR?
F08C C2  9E  F0     1320    JNZ ALPL
F08F 2F             1330 MODESW CMA
F090 E6  04         1340    ANI 4
F092 32  D9  F1     1370    STA MODFL
F095 CD  00  F0     1380 SLOOP CALL STATUS
F098 CA  95  F0     1390    JZ SLOOP
F09B C3  59  F0     1410    JMP PS
F09E 21  98  F1     1420 ALPL LXI H,ALOOK        BAUDOT LOOKUP TBL
F0A1 3A  D9  F1     1430    LDA MODFL
F0A4 B7             1440    ORA A                SET FLAGS
```

Totally Integrated, Entirely Self-Contained

THE PET™

PERSONAL COMPUTER

With technology so advanced,
Concept so remarkable,
Operation so utterly simple,
Cost so incredibly low.
The PET has given rise to a brand new era...
The Age of the Personal Computer

HIGH SPEED PRINTER ACCESSORY

Immediate Delivery

FEATURING AN IEEE-488 BUS

THE PET has become the standard for the personal computer industry. Consumer and business publications have lauded its discovery. POPULAR SCIENCE and PLAYBOY have given special tribute to the "mind-boggling" PET.

IN A LEAGUE WITH IBM, HP AND WANG MINICOMPUTERS

THE PET is a minicomputer and should not be confused with game products that hook up to household T.V.'s. **What sets it apart from other computers is price. While others cost from $11,000 to $20,000 and more, THE PET, with similar power, costs only $795.00.**

Features an IEEE-488 Bus – like HP's mini and full size computers. This standard data and control channel permits direct connection to many peripherals. Over 120 pieces of compatible equipment such as counters, timers, spectrum analyzers, digital voltmeters and printer plotters, from HP, Philips, Fluke, and Textronix, etc., are currently available.

ROM Magazine, January 1978, writes, "THE PET comes out of the box, plugs into the wall, and is ready to use." It is equipped with a CRT video display with reverse and blink features, an alpha-numeric keyboard with complete graphics and a built-in standard cassette tape deck.

THE PET has 8K bytes of RAM (user memory). Optional equipment permits expansion to 32K. And, it has 14K bytes of ROM (program memory).

THE PET COMMUNICATES IN BASIC. THE EASIEST COMPUTER LANGUAGE

If THE PET wants you to press a key, it will flash, "Press such and such", on the display. You speak back to it through its full size 73-key keyboard.

EXTENSIVE CHARACTER ORIENTED GRAPHICS

The unit features a 9-inch, high resolution, 1000 character CRT. Characters are arranged 40 columns by 25 lines on an 8 x 8 matrix for superb graphics.

WHAT IS THE PET REALLY FOR?

It is the single most important teaching device for any computer related subject. It will entertain the most sophisticated data application, or the simplest inquiry/response assignment. **IN THE LAB** it handles instrumentation, process monitoring, and more. A number of Fortune 500 companies have already made it an integral part of their lab and general office system.

As a **BUSINESS TOOL** it will; Maintain ledgers. Keep payroll records. Create P & L's. Control inventory. Store and analyze sales data. Draw bar graphs. Issue invoices. Hook up to on-line computer system. **AT-HOME** it will; Compute state and federal tax returns, Make heat and insulation analyses. Keep Christmas lists. Keep checkbook and finances up to date. A variety of games, from Blackjack to Galaxy, is currently available.

Bar Graphs

Amortization Chart

Black Jack

Teaching Trigonometry

HIGH SPEED PET PRINTER

This powerful word processor prints hardcopies, invoices, computer correspondence. Faster than an IBM Selectric, THE PET Printer delivers 60 characters per second at a sustained rate – with upper and lower case capability. Characters are one-eighth inch tall and are printed in a 7 x 8 dot matrix. The printer uses a standard 8½" wide paper roll. And, it is only $695.00.

PERIPHERAL SECOND CASSETTE

This optional component expands storage and increases flexibility. Only $99.95.

MILES OF SOFTWARE

Many programs are available now, including, "BASIC BASIC" which shows how to write a program. You can develop your own programs to meet personal requirements.

TECHNICAL SPECIFICATIONS

MEMORY
Random Access Memory (user memory), 8K internal, expandable to 32K bytes
Read Only Memory (operating system resident in the computer), 14K bytes
8K-BASIC interpreter program, 4K-Operating system, 1K-Diagnostic routine

VIDEO DISPLAY UNIT
9" enclosed, black & white, high resolution CRT
1000 character display, arranged 40 columns by 25 lines
8 x 8 dot matrix for characters and continuous graphics
Automatic scrolling from bottom of screen
Winking cursor with full motion control
Reverse field on all characters
64 standard ASCII characters; 64 graphic characters

KEYBOARD
9½" wide x 3" deep, 73 keys
All 64 ASCII characters available without shift.
Calculator style numeric key pad
All 64 graphic and reverse field characters accessible from keyboard (with shift)
Screen Control: Clear and erase
Editing: Character insertion and deletion

CASSETTE STORAGE
Fast Commodore designed redundant-recording scheme, assuring reliable data recovery
Cassette drive modified by Commodore for much higher reliability of recording and record retention
High noise immunity, error detection, and correction
Uses standard audio cassette tapes
Tape files, named

OPERATING SYSTEM
Supports multiple languages (BASIC resident)
Machine language accessibility
File management in operating system
Cursor control, reverse field, and graphics under simple BASIC control
Cassette file management from BASIC
True random number generation or pseudo random sequence

INPUT/OUTPUT
All other I/O supported through IEEE-488 instrument interface for peripherals
I/O automatically managed by operating system software
Single character I/O with GET command
Easy screen line-edit capability
Flexible I/O structure for BASIC expansion with peripherals

BASIC INTERPRETER
8K BASIC; 20% faster than most other 8K BASICS
Upward expansion from BASIC language
Strings, integers, multiple dimension arrays
10 significant digits; floating point
Direct memory access: PEEK and POKE commands

DIMENSIONS
16" wide: 18½" deep; 14" high. Weight: 44 lbs.

GAME PROGRAMS ARE $9.95 EACH:
☐ **Black Jack** ☐ **Draw Poker** ☐ **Galaxy Games**
☐ **Space Flight** ☐ **Target Bong, Off-The-Wall**
☐ **Lunar Lander, Wumpus, Rotate, Tic-Tac-Toe**
☐ **Osero, Reverse** ☐ **Spacetrek** ☐ **Kingdom**

PROGRAMS AT $14.95 EACH:
☐ **Mortgage Analysis**
☐ **Diet Planner and Biorhythm**
☐ **Basic Basic**-by Lodewyck and James

PROGRAMS AT $24.95 EACH:
☐ **Basic Investment Analysis**-loans, annuities, return on regular and irregular sequences of payments, calendar calculations
☐ **Stock Portfolio Recordkeeping and Analysis**-keeps track of buys, sells, and dividends. Calculates current value, rates of return
☐ **Checkbook Recordkeeping and Analysis**-keeps track of checks and deposits. Analyzes expenses by date and type

PROGRAMS AT $29.95 EACH:
☐ **Basic Math Package**-matrix addition, multiplication, determinants and inverses to 16 x 16, solution of simultaneous linear equations, vector and plane geometry calculations, integration by trapezoidal, Simpson's rule or Gaussien quadrature, differentiation
☐ **Basic Statistics Package**-mean, median, variance, standard deviation, skewness, kurtosis, frequency distribution, linear regression, T-tests, correlation analyses

FREE ORIENTATION PACKAGE

Your PET comes complete with two programs and an easy-to-follow instruction manual. By working through the routines you will quickly discover how easy it is to gain command of your personal computer.

SERVICE WORLDWIDE

Because your PET is self-contained and compact, professional factory service is never far away. If major service is required, the unit can simply be returned by UPS to an authorized Commodore PET clinic.

To order your PET send check or money order for $795.00 plus $20.00 for shipping and insurance. To order the PET Printer, add $695.00 plus $12.00 for shipping and insurance. The Second Cassette is $99.95. No shipping and insurance charges are *required* when ordering a second cassette or programs with your PET. Credit card orders are invited to call our toll free number below. Orders will be accepted on our TELEX, No. 25-5268.

Use THE PET for 30 days with no obligation. If, for any reason, you are not satisfied, return it for a prompt and courteous refund.

ORDER DIRECT

CREDIT CARD ORDERS CALL TOLL FREE

800-323-2272

ILLINOIS RESIDENTS CALL: 312-595-0461
TELEX ORDERS: 25-5268

Order your PET, Printer Accessory, Second Cassette and Programs from Contemporary Marketing at:

790 MAPLE LANE DEPT. KI-9
BENSENVILLE, IIIINOIS 60106 C65

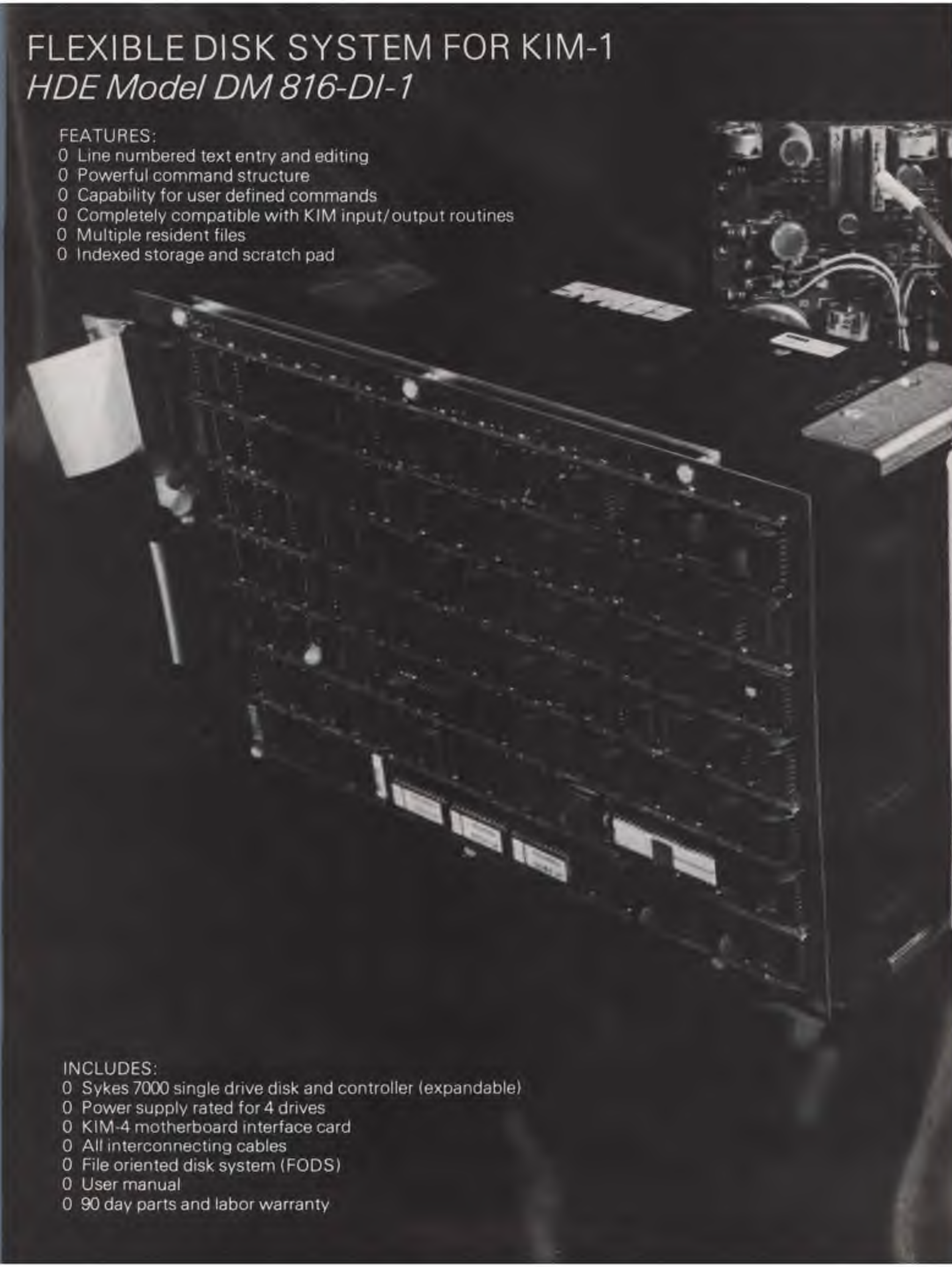
Contemporary Marketing Inc.

```
F0A5 CA    AB   F0     1450     JZ LOOKUP
F0A8 21    B8   F1     1460     LXI H,FIGLOOK
F0AB 09                1470 LOOKUP DAD B
F0AC 3E    01          1480     MVI A,1
F0AE CD    4E   F0     1481     CALL OUTSHIFT
F0B1 01    44   48     1483     LXI B,18500
F0B4 7E                1484     MOV A,M
F0B5 FE    0D          1485     CPI ODH
F0B7 CC    90   F1     1486     CZ LOOP2
F0BA 7E                1487     MOV A,M
F0BB 21    DA   F1     1490     LXI H,SUPFL
F0BE 36    01          1510     MVI M,1
F0C0 C1                1522 INRET POP B
F0C1 E1                1530     POP H
F0C2 D1                1540     POP D
F0C3 C9                1550     RET
F0C4                   1560 X
F0C4                   1570 X
F0C4                   1580 X
F0C4                   1805 X
F0C4                   1900 X
F0C4                   1951 X
F0C4                   1952 X -------------------------------------
F0C4                   1953 X
F0C4                   1954 X  OUTPUT ROUTINE
F0C4                   1955 X        - CONVERTS THE ASCII CODE IN
F0C4                   1956 X          ACCUMULATOR TO SERIAL BAUDOT
F0C4                   1960 X          CODE TO DRIVE THE BAUDOT
F0C4                   1962 X          TELEPRINTER
F0C4                   1964 X
F0C4                   1965 X -------------------------------------
F0C4                   1966 X
F0C4 FE    3E          2000 OUTPUT CPI 3EH           ASCII GT
F0C6 C2    D1   F0     2010     JNZ TRYLT
F0C9 3E    47          2020     MVI A,'G'
F0CB CD    EC   F0     2030     CALL OUT
F0CE C3    DB   F0     2040     JMP THAN
F0D1 FE    3C          2050 TRYLT CPI '<'
F0D3 C2    E0   F0     2060     JNZ TRYEQ
F0D6 3E    4C          2070     MVI A,'L'
F0D8 CD    EC   F0     2080     CALL OUT
F0DB 3E    54          2090 THAN MVI A,'T'
F0DD C3    EC   F0     2110     JMP OUT
F0E0 FE    3D          2120 TRYEQ CPI '='
F0E2 C2    EC   F0     2130     JNZ OUT
F0E5 3E    45          2140     MVI A,'E'
F0E7 CD    EC   F0     2150     CALL OUT
F0EA 3E    51          2155     MVI A,'Q'
F0EC D5                2160 OUT PUSH D
F0ED E5                2170     PUSH H
F0EE C5                2180     PUSH B
F0EF F5                2190     PUSH PSW
F0F0 E6    7F          2210     ANI 7FH                REMOVE PARITY
F0F2 FE    60          2220     CPI 60H                CONVERT LC TO UC
F0F4 DA    F9   F0     2230     JC UPCASE
F0F7 D6    20          2240     SUI 20H
F0F9 FE    58          2250 UPCASE CPI 'X'
F0FB C2    00   F1     2252     JNZ UPC2
F0FE 3E    2A          2254     MVI A,'X'
F100 5F                2256 UPC2 MOV E,A             SAVE IN E
F101 21    DA   F1     2260     LXI H,SUPFL
F104 AF                2270     XRA A
F105 BE                2280     CMP M                  MEQO ?
F106 77                2290     MOV M,A                CLEAR FLAG
F107 C2    6D   F1     2310     JNZ OUTRET             NO OUTPUT
```

```
F10A 21    98   F1     2320     LXI H,ALOOK            START OF TABLE
F10D 16    40          2330     MVI D,64               COUNTER
F10F 7B                2340     MOV A,E                PUT ASCII IN A
F110 BE                2350 LUP CMP M
F111 CA    1C   F1     2360     JZ MATCH
F114 23                2370     INX H
F115 15                2380     DCR D
F116 C2    10   F1     2390     JNZ LUP
F119 C3    6D   F1     2410     JMP OUTRET             NOT IN TABLE
F11C E5                2420 MATCH PUSH H             POINTER
F11D FE    0D          2430     CPI ODH                ASCII CR
F11F CA    50   F1     2440     JZ SERIAL
F122 FE    0A          2450     CPI OAH                ASCII LF
F124 CA    50   F1     2460     JZ SERIAL
F127 FE    20          2470     CPI ' '
F129 CA    50   F1     2480     JZ SERIAL
F12C 3E    20          2490     MVI A,32
F12E BA                2520     CMP D                  FIG IF D=1 TO 32
F12F 21    D9   F1     2530     LXI H,MODFL
F132 3E    00          2540     MVI A,0
F134 D2    45   F1     2550     JNC FIG
F137 BE                2560     CMP M
F138 36    00          2570     MVI M,0                RESET MODE SWITCH
F13A CA    50   F1     2580     JZ SERIAL
F13D 3E    1F          2590     MVI A,1FH              BAUDOT LETTER KEY
F13F CD    75   F1     2610     CALL TTYOUT
F142 C3    50   F1     2620     JMP SERIAL
F145 BE                2630 FIG CMP M
F146 36    04          2640     MVI M,4
F148 C2    50   F1     2650     JNZ SERIAL
F14B 3E    1B          2660     MVI A,1BH              BAUDOT FIG KEY
F14D CD    75   F1     2670     CALL TTYOUT
F150 E1                2680 SERIAL POP H             POINTER
F151 F5                2685     PUSH PSW
F152 01    98   F1     2690     LXI B,ALOOK            LOOK UP TABLE
F155 7D                2710     MOV A,L                A = ADRESS +
F156 91                2720     SUB C                          TTY CODE
F157 DE    20          2730     SBI 32
F159 F2    5E   F1     2740     JP OKB
F15C C6    20          2750     ADI 32
F15E CD    75   F1     2760 OKB CALL TTYOUT
F161 F1                2762     POP PSW
F162 FE    0D          2764     CPI ODH                ASCII CR
F164 C2    6D   F1     2766     JNZ OUTRET
F167 01    44   48     2768     LXI B,18500            DELAY 1/4 SEC
F16A CD    90   F1     2770     CALL LOOP2
F16D F1                2775 OUTRET POP PSW
F16E CD    75   FF     2776     CALL VDMOUT            OPTIONAL FOR VIDEO
F171 C1                2780     POP B
F172 E1                2790     POP H
F173 D1                2810     POP D
F174 C9                2830     RET
F175 07                2900 TTYOUT RLC               ROTATE LEFT
F176 E6    FE          2910     ANI 0FEH               BIT 0 EQ START BIT
F178 F6    C0          2920     ORI 0C0H               BITS 6,7 EQ STOP BITS
F17A 16    08          2930     MVI D,8                D EQ COUNTER
F17C 07                2940     RLC
F17D 0F                2950 TLOOP RRC
F17E 5F                2960     MOV E,A
F17F E6    01          2970     ANI 1
F181 CD    4E   F0     2980     CALL OUTSHIFT
F184 CD    8D   F1     2990     CALL DELAY
F187 7B                3010     MOV A,E
F188 15                3020     DCR D
F189 C2    7D   F1     3030     JNZ TLOOP
```

```
F18C C9              3040     RET
F18D 01   59   06    3100 DELAY LXI B,1625        22 MS DELAY
F190 0B              3120 LOOP2 DCX B
F191 78              3130     MOV A,B
F192 FE   FF         3140     CPI 0FFH
F194 C2   90   F1    3150     JNZ LOOP2
F197 C9              3160     RET
F198                 3170 X
F198                 3172 X  LOOK UP TABLE FOR LETTERS
F198                 3174 X
F198 03   45         4000 ALOOK DW 'ED'-40H
F19A 0A   41         4010     DW 410AH
F19C 20   53         4020     DW 'S '
F19E 49   55         4030     DW 'UI'
F1A0 0D   44         4040     DW 440DH
F1A2 52   4A         4050     DW 'JR'
F1A4 4E   46         4060     DW 'FN'
F1A6 43   4B         4070     DW 'KC'
F1A8 54   5A         4080     DW 'ZT'
F1AA 4C   57         4090     DW 'WL'
F1AC 48   59         4110     DW 'YH'
F1AE 50   51         4120     DW 'QP'
F1B0 4F   42         4130     DW 'BO'
F1B2 47   20         4140     DW ' G'
F1B4 4D   2A         4150     DW 'XM'
F1B6 56   20         4160     DW ' V'
F1B8                 4162 X
F1B8                 4164 X  LOOK UP TABLE FOR FIGURES
F1B8                 4166 X
F1B8 7F   33         4170 FIGLOOK DW 337FH
F1BA 0A   2D         4180     DW 2D0AH
F1BC 20   27         4190     DW 2720H
F1BE 38   37         4210     DW '78'
F1C0 0D   24         4220     DW 240DH
F1C2 34   07         4230     DW 0734H
F1C4 2C   21         4240     DW '~,'
F1C6 3A   28         4250     DW '(:'
F1C8 35   22         4260     DW '"5'
F1CA 29   32         4270     DW '2)'
F1CC 23   36         4280     DW '6 #'
F1CE 30   31         4290     DW '10'
F1D0 39   3F         4310     DW '?9'
F1D2 2B   20         4320     DW ' &'
F1D4 2E   2F         4330     DW '/.'
F1D6 3B   20         4340     DW ' ;'
F1D8                 4350 X
F1D8                 4352 X  THE FOLLOWING SOFTWARE FLAGS
F1D8                 4354 X  MUST BE LOCATED IN RAM
F1D8                 4356 X
F1D8                 4500 PORTSEL DS 1
F1D9 00              4510 MODFL DB 0
F1DA 00              4520 SUPFL DB 0
F1DB                 4600 X
F1DB                 4602 X SYSTEM EQUATES
F1DB                 4604 X
F1DB                 5000 KBDSTATUS EQU 0
F1DB                 5010 BIT EQU 1
F1DB                 5020 IPORT EQU 6EH           TARBELL SPARE
F1DB                 5030 OUTPORT EQU 6EH         TARBELL SPARE
F1DB                 5040 KBD EQU 1
F1DB                 5050 PSW EQU 6
F1DB                 5090 X
F1DB                 5095 X
F1DB                 5096 X
F1DB                 6000 X
```

DEALER DIRECTORY



FLEXIBLE DISK SYSTEM FOR KIM-1
HDE Model DM 816-DI-1
FEATURES:
0 Line numbered text entry and editing
0 Powerful command structure
0 Capability for user defined commands
0 Completely compatible with KIM input/output routines
0 Multiple resident files
0 Indexed storage and scratch pad
INCLUDES:
0 Sykes 7000 single drive disk and controller (expandable)
0 Power supply rated for 4 drives
0 KIM-4 motherboard interface card
0 All interconnecting cables
0 File oriented disk system (FODS)
0 User manual
0 90 day parts and labor warranty

Gary J. Shannon
9856 Haskell Ave.
Sepulveda CA 91344

Error-Correcting Techniques

Reliability has been a goal in data transmission ever since the earliest telegraph lines were stretched across the country. The following discusses some applications a little more up-to-date than that.

Next to speed, the most common complaint about the cassette interfaces available to the hobbyist today is their error rate. Nothing is more frustrating than losing a favorite program or an important piece of data simply because one bit out of several hundred thousand on the tape was lost to a wrinkle or chip in the oxide coating of the tape.

The most obvious solution is to keep several backup copies of important tapes on hand. A simpler, and often more satisfactory, solution is to make three or four copies of the program, one after another, on the same tape. This safeguard is all but standard in the world of commercial data processing and is the surest protection against hard errors due to physical damage to the tape, such as breakage, chipping or spilled coffee.

But there is another kind of error which, though less serious, is more annoying because it happens more often. This is the so-called soft error—the tape reads perfectly eight times in a row, but then fails to read when you try to show off your system to a friend. This prompts remarks such as, "Do you always have to try it three times before it works?"

This type of error is caused by some fleeting and unknown gremlin in the system that, now and then, likes to waylay a bit at random. You can be reasonably sure that it is, indeed, a soft error if a second or third try produces a clean, error-free load.

Most cassette save/load software recognizes the possibility of error, and provides what is called the checksum to verify that the data has, in fact, been loaded correctly. Basically, the checksum is the binary sum of every byte, ignoring all carry bits, on the tape. If the checksum computed during a load operation agrees with the checksum read off the tape, the odds are 256 to 1 that the load is correct.

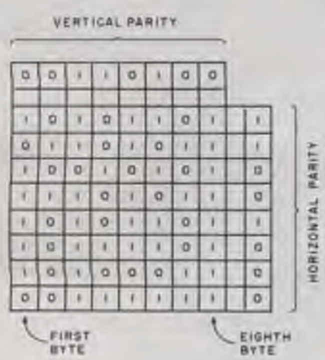


Fig. 1. A block of eight bytes showing vertical and horizontal parity.

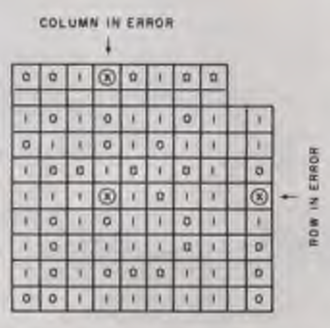


Fig. 2. Two-way parity allows us to zero in on the bit in error.

Error Correction Is the Answer

If we want completely trouble-free operation, however, it is not enough that the computer tell us that some error has occurred; the system should go beyond that and actually *correct* the error. If this were possible, there would be no need for the program to even tell us that an error had happened. It could simply straighten out the problem, fix up the bad data and keep right on going without ever missing a beat! From the user's point of view, a tape-cassette system that could repair its own errors "on the fly" would look the same as a cassette that never made any errors in the first place.

The tape transports used on large-scale computers are many times more reliable than the cassette deck used by the average hobbyist, but even these have elaborate error-detection and correction circuitry built into them. You may consider it a personal tragedy when you lose your favorite version of Star Trek, but consider the consequences if that tape contained the entire company's payroll or a few thousand checking-account balances. The large-scale tape transports simply must work. There is very little margin for error.

Does this mean that in order to achieve a high level of reliability we must reengineer our cassette interface boards? Not at all. What every large-scale commercial system accomplishes with exotic hardware, we can do quite effectively in software. That's right, your existing cassette interface can be made 10, or 100, or even 1000 times more reliable with appropriate error-correcting software.

"But hold on!" you say. "You can't get something for nothing. What exactly must we give up to get that kind of reliability?" The answer is speed. The system of error correction we will explore here produces

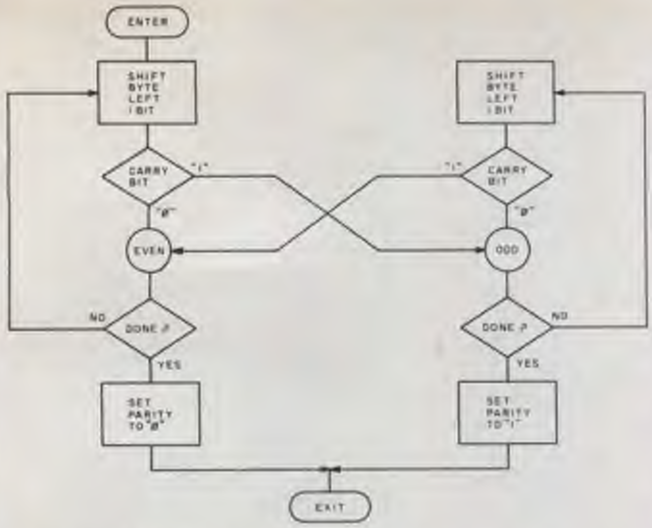


Fig. 4. A software flip-flop computes vertical parity bits.

about 16 times the reliability at a cost of about 20 percent reduction in speed. Other systems that can boost reliability by many hundreds of times may reduce speed by as much as two-thirds.

Parity—What Is It?

Before getting down to specifics, let's consider the concept of parity. In its simplest sense, parity is a single-bit checksum computed for a single byte. In other words, it takes the binary sum of all the ones and zeros in the byte, and discards all but the rightmost bit.

In some systems, parity is the complement of this sum; such buzzwords as "odd parity" or "even parity" are used to describe these differences. But the parity bit, like the checksum, only tells us that an error has occurred; it does not provide any clue as to which bit was in error.

Fig. 3. The vertical parity can be written as a separate byte at the end of the data block.

Consider, for example, the byte 0110 0011. The single-bit sum of all the bits is 0. Now suppose that an error occurred while this byte was being read off the tape, and it was received as 0111 0011. The computed parity would be 1, but the parity we were told to expect was 0. We know that one of the bits we read in was in error, but which one? It might even be the parity bit itself.

Clearly, parity, by itself, will never give the computer all the information it needs to *correct* a bit in error. In order to correct errors, we might use a system similar to that used in most large-scale systems. In these systems we usually find two kinds of parity—vertical and horizontal parity, or row and column parity.

Imagine that we have a block of eight bytes written on tape, as in Fig. 1. If we think of the bytes as occupying vertical columns, set side by side, we can easily see how these two different kinds of parity would be computed.

The so-called vertical parity bit, the single-bit checksum for each bit in the byte, is the parity we have been discussing. The horizontal parity, on the other hand, is the same computation made horizontally—on a single-bit position in each of the eight bytes in the block. Although the data is not actually arranged in this manner on the tape (except in large-scale nine-track recording systems), the data can be effectively treated as if it were.

Now suppose that after we read in the entire block, the computed parity for byte 6 disagrees with the input parity. Some error has occurred in byte number 6, but in which bit? Now we must examine the horizontal bits. If any bit of byte 6 were changed, a parity error would show up in the horizontal row corresponding to the position of that bit. The presence of two simultaneous parity errors, row and column, allows us to zero in on the exact bit that was received incorrectly, as illustrated in Fig. 2.

If we know that a bit read in as 1 is in error, then, since the only possible values are 1 and 0, the only possible error is that the 1 should be a 0. By the same reasoning, if the bit in error had been a 0, then it should be changed to a 1 to correct the error. In other words, once the bit in error has been located,the correction is simply to complement that bit.

What about Hobby Systems?

This all sounds great in theory, but the cassette interface is usually set up to transmit and receive data in eight-bit bytes. How do we coax such a device to write the ninth bit we need for parity? The simplest solution is to save all the parity bits for a given block and put them out as a single extra parity *byte.*

Fig. 3 illustrates how this arrangement works. A block of eight data bytes now becomes a block of ten bytes—eight data bytes, one horizontal parity byte, and one vertical parity byte. Now we see why increased reliability means lowered speed; the extra bytes require extra time to read and write.

The actual coding required to implement this method of error correction depends, of course, on which processor and which cassette interface is being used. Rather than try to provide examples of every possible combination, we will consider a detailed description of the logic required. I hope this will be sufficient information for the average assembly-language programmer who wishes to implement the software on his own system.

The horizontal parity is most easily formed by using the exclusive OR instruction. Each byte to be written is XORd into the byte that will become the horizontal parity byte. This has the effect of placing a zero wherever an even number of bits were ones, and a one wherever an odd number of bits were ones. The vertical parity byte requires a little more logic to generate.

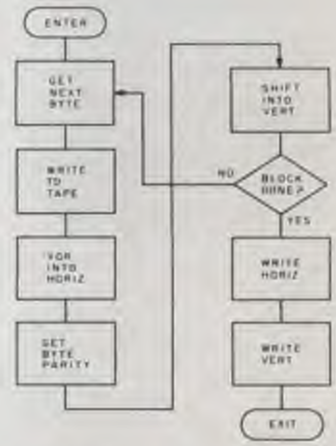


Fig. 5. The procedure for writing a self-correcting data block.

The most significant bit in the parity byte is the parity for the most significant byte of the block, and so on. In order to accomplish this, each parity bit must be generated, and then shifted into the prior intermediate parity byte until all eight bytes have been written and *all eight* parity bits have been shifted into the parity byte.

The method used to *generate* the individual parity bits will vary from one processor to another, and, in fact, some machines compute the parity bit automatically. If yours is one that does not, consider the simple program illustrated in Fig. 4 to carry out the necessary computation. The subroutine is essentially a two-state automaton, or software flip-flop, that changes state (from odd to even or vice versa) whenever a one bit is encountered. The

subroutine is entered in the even state and exits from the appropriate state when all eight bits have been processed. The parity bit is determined by the final state of the subroutine.

The procedure for writing a block is diagramed in Fig. 5. The flowchart of Fig. 6 is used to read back the same block with error correction.

It should be noted that certain types of errors cannot be corrected by this method. They include multiple errors in a single block, and single errors in the parity bytes themselves. Greater reliability could be provided by the addition of a third control byte to contain the parity for the two prior parity bytes. In case an error occurs in one of the parity bytes, the third control byte would indicate the presence of an error; if the other parity byte were correct, we would know that the error was in the parity byte only, and that the data is correct and needs no fixing. This would probably eliminate most errors that would otherwise be flagged as multibit errors.

Wrap-up

Given a piece of equipment with fixed reliability, more dependable operation can be obtained only through increased redundancy. However, as redundancy increases, speed decreases.

One very simple technique with far greater reliability than the scheme discussed above is to write each byte three times in a row and then apply a majority-rules vote among the three incoming bytes to decide which bits should be ones. It is extremely unlikely that three bytes in a row would suffer from the same error; the majority vote of the other two bytes would ignore the oddball if an error does come up.

This simple trick increases reliability by several hundred times, but slows the effective transfer rate to one-third the normal rate for the interface.

In the final analysis, the choice of technique will rest solely on the relative importance given to the two factors of speed versus reliability. The choice is up to you.■

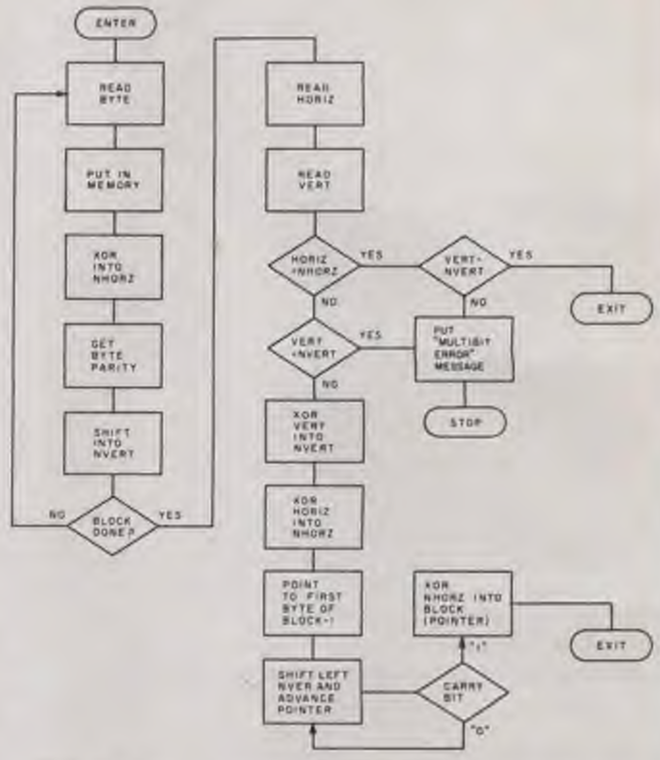


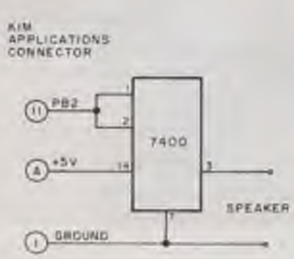
Fig. 6. Retrieving the data block with error correction.

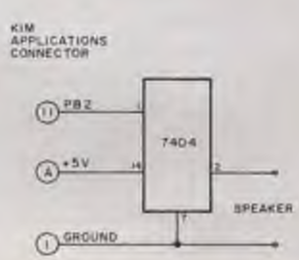
KIM Organ

Short . . . but sweet notes will emanate from your KIM.

Don Ketchum
313 Van Ness Ave.
Upland CA 91786

The following program and either of the two simple interface circuits will turn your KIM-1 into a 15-note electronic organ, playable directly from the KIM keyboard.■





KEY	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
NOTE	F	G	A	B	C	D	E	F	G	A	B	C	D	E	F

ADDRESS	PROGRAM							
100	D8	A2	04	20	1D	01	A2	00
108	20	1D	01	20	6A	1F	C9	15
110	F0	F9	A8	B9	35	01	8D	2B
118	01	D0	E6	00	00	8E	03	17
120	A2	41	8E	1B	01	CE	1B	01
128	D0	FB	A2	00	8E	1C	01	CE
130	1C	01	D0	FB	60	FA	FA	D6
138	B6	96	8A	74	5E	55	42	33
140	24	1C	10	07	01	01	01	01
148	01	01						

FIRST CLASS
Permit No. 17
Peterborough NH 03458

BUSINESS REPLY MAIL

No postage stamp necessary if mailed in the United States

Postage will be paid by

kilobaud

Peterborough NH 03458

MICROCOMPUTER MADNESS*

*OR YOUR MONEY BACK!

(use coupon below and your enjoyment is guaranteed)

224 ARTICLES

Kilobaud has more articles than any other microcomputer magazine. During 1977, for instance, there were 224 articles in Kilobaud—880 pages of articles—a veritable encyclopedia of microcomputing. There will be even more articles in 1978.

FOR THE BEGINNER

Though Kilobaud covers both the technical and programming sides of microcomputers, each article is written to be understandable for the beginner. No other magazine makes it as easy to learn more about microcomputers.

THE BEST PROGRAMS

Only Kilobaud offers programmers both the publication of their programs in the magazine plus a large royalty for the program if it's issued on cassettes. Is it any wonder all of the really good programs are being published in Kilobaud? You'll find the best in software . . . games, diagnostics, teaching, music, business, etc. We're very heavy on programs.

HOBBY OR BUSINESS?

Both businessmen and hobbyists want the same thing: to understand microcomputers . . . and this is the purpose of Kilobaud. KB also has articles aimed at the businessman to help him know what is available and what it will do.

kilobaud

Understandable for beginners . . . Interesting for experts.

peterborough nh 03458

MONTHLY COLUMNS

Keep up with the latest developments and readers' programs through the letters column. The editorials will keep you up to date on money-making ideas, the progress of the industry and more. The New Products column is particularly useful, including the results of tests of the latest equipment in the Kilobaud microcomputer laboratory—the most complete in the industry.

SUBSCRIBE TO KILOBAUD

Since Kilobaud is the fastest selling microcomputer magazine on the newsstand, it's sometimes hard to find a copy. Why not subscribe today and get the convenience of mail delivery, plus a big cash savings . . . $6 over newsstand for a 1-year subscription and a $36 savings when you subscribe for 3 years. If you are in a hurry, call Toll Free 1-800-258-5473, and enter your subscription right away . . . we'll bill you later.

New from Kilobaud: INSTANT SOFTWARE
—See page 115

KILOBAUD

100% GUARANTEE OF EXTREME SATISFACTION!

That's right, if after trying KILOBAUD for 60 days you are willing to cancel your subscription, we'll refund your money begrudgingly. (We work 18 hours a day, seven days a week, to put out the best microcomputer magazine the world has ever seen, jam-packed with great articles and fantastic programs and won't pretend to be happy about being rejected) Your check will be in the next day's mail.

☐ 1 year $18 ☐ 3 years $36 ☐ Bill me ☐ Payment enclosed

Ms./Mr. (please print)

Address

City State Zip

For Instant Subscriptions call 1-800-258-5473 or write to KILOBAUD / Peterborough NH 03458

KILOBAUD KLASSROOM NO. 12

ROM and RAM Memories

Before we can go ahead and build our computer, we need to learn more about memories.

Peter A. Stark
PO Box 209
Mt. Kisco NY 10549

In the last session, we finished up the data and address buses and covered address decoding for memory.

In this session, we will look at ROM and RAM memories and how they work.

Introduction

The most common memory is called RAM, which stands for random access memory. Read-write memory, or RWM, might be a better choice, but it doesn't pronounce as well. RAM is memory that can be written into or read out of by the computer at any time, and in any order. The last part is what is meant by the word random. Every computer will need some RAM, and usually the more the better.

ROM, or read only memory, can also be written and read, but the writing is usually done only once or at best a small number of times. ROM comes in several types.

1. Mask-programmed ROM is written into by the manufacturer. Actually, the pattern of 0 and 1 data is permanently stored into it by the IC manufacturer during the time that the chip itself is being made. Mask-programmed ROMs are cheap in large quantities, but the setup charge of $1000 and up makes it impractical to have just one or two made . . . definitely only for the big spenders.

2. PROM is programmable ROM. The PROM is manufactured containing either all 0s or all 1s, depending on the IC, and it can be changed at a later time by applying higher voltage or current pulses to the pins in a specified sequence. Most PROMs use fusible links, very thin interconnecting links on the IC chip which burn up and open—like a fuse—when zapped by a high-current pulse. If the PROM initially contains all 1s, then "burning" the PROM opens the links and changes them to 0s. Once programmed, additional 1s can still be changed to 0s, but the 0s cannot be changed back to 1s.

3. EPROM is also a programmable ROM, but it is erasable. Rather than burning up fusible links, the EPROM stores its data as electron charges in isolated regions on the semiconductor. Once a charge is placed in a particular bit, it cannot leak off since the region in which it is located is insulated from the rest of the IC. IC manufacturers claim that the memory will be retained for at least ten years, though occasionally an EPROM does seem to have a short memory. The EPROM can be erased by placing it in very strong ultraviolet light. This shortwave high-energy light makes the EPROM chip act like a photodetector by increasing the leakage currents in the silicon material. Given enough time, this can discharge the stored charges and erase the EPROM.

4. EAROM is an electrically alterable ROM. It is actually quite similar to an EPROM, except that the erasing is done with an applied voltage rather than applied light. Thus, the erasing can be done while the IC is still plugged into the computer, and the computer can write back into it. Typical read times are around one microsecond or less, whereas erasing and writing times are on the order of one millisecond or more; so writing takes much longer than reading. For this reason, writing would be done very seldom; another name you sometimes hear for the EAROM is RMM, or read mostly memory.

The ROM makes a good beginning to the study of memories, since it is easy to build and understand.

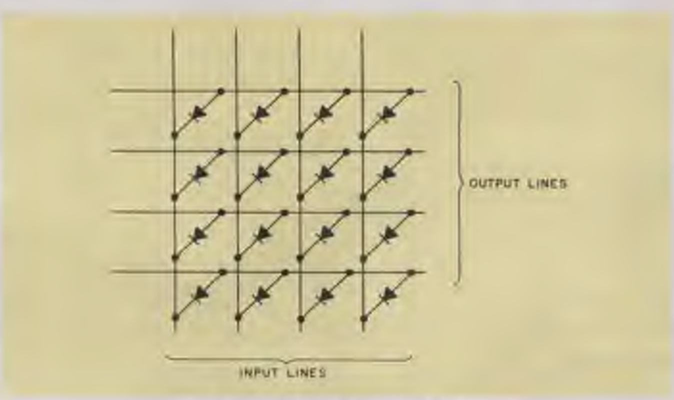


Fig. 1. Simple diode ROM.

Experiment #56
A Simple Diode ROM

Problem: What is the simplest kind of a ROM?

Solution: Diode ROMs have been used in a variety of circuits for many years, even before the coming of integrated circuits. We will build one on the console breadboard.

Procedure: Fig. 1 shows the ROM. In this case, we have 16 diodes arranged in a neat matrix of four across and four down. This kind of a circuit could be built fairly neatly on a two-sided printed circuit board, with the vertical input lines running on one side of the board and the horizontal output lines on the other. On the breadboarding socket of your console, however, it is much harder to wire up in an orderly manner. Any way you try to do it, you will wind up with a rat's nest of wires.

For this experiment, use the junkiest diodes you can find. Use small signal diodes—

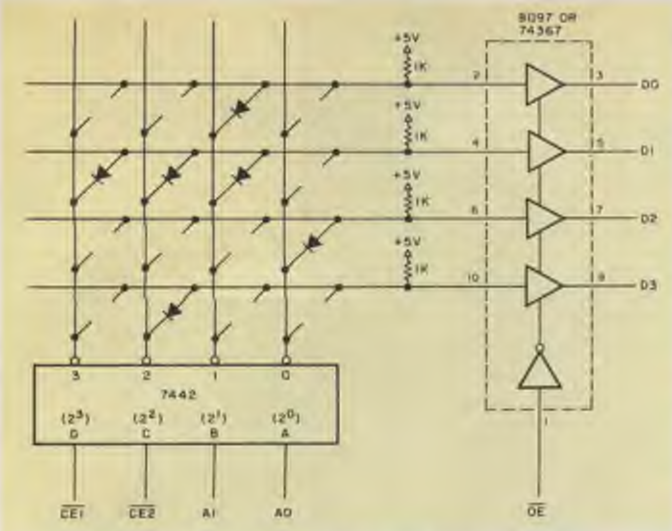


Fig. 2. Homemade ROM using a diode matrix.

either germanium or silicon will do—the smaller the better. Do not use big power rectifier diodes.

Next, you will need a way of monitoring the output of the ROM. To each output line hook up an LED and a 150 Ohm resistor in series to +5 volts; make sure that the anode of the diode goes up toward the +5 volt line. At this point, the LEDs will not light yet.

Now take a short wire jumper and ground any one of the four input lines. Grounding any one of these lines should turn on all four output LEDs. Ground each input line in turn to make sure that each one turns on all the LEDs. If any one of the LEDs does not light when one of the input lines is grounded, then probably the diode at the intersection of the grounded input line and the line to the dark LED is defective or wired wrong or backward.

Theory: When none of the input lines are grounded, the output lines are floating, and all four output LEDs are off. This is equivalent to a 1 output. If we ground one of the input lines, we complete the circuit to ground, and each of the output lines is pulled to ground through a diode.

This would be called a 4 x 4 ROM, meaning that it has four memory locations, each having four bits. Each of the vertical input lines corresponds to one memory location, while the four horizontal output lines carry one bit each. At this point, each of the four locations is programmed with all 0s, since all outputs go to a 0 or low voltage when one of the locations is selected by grounding an input line; but if the output were connected to inverters then it would look as though all locations were programmed to high voltages or 1s.

Experiment #57
Programming the Diode ROM

Problem: How do we program such a ROM?

Solution: By burning out diodes or by unplugging them from the socket. Actually, it depends on your mood . . . you may decide to simply unplug them. On the other hand, if you want a thrill, and hang the cost, then proceed with the rest of this experiment.

Procedure: Assuming that you used really cheap diodes, and that your +5 volt power supply has enough current capacity, you should have no problem burning out a diode at a time. Simply ground an input line with a short jumper and then connect an output line to +5 volts with another jumper. ZAP . . . with luck you'll see a slight flash inside the diode . . . and there it is. You have just changed a 0 into a 1.

But watch out. It is possible for a glass diode to shatter, so keep it far enough away from your face. Also, do not try this if you have used rectifier diodes in the diode ROM. You might find that you are blowing out your power supply instead of just a ROM diode.

After you have burned—or unplugged—a few diodes, remove the jumper to +5 volts and again ground each of the four input lines, one at a time. This time only those LEDs connected to good diodes will light. In other words, good diodes still provide 0 outputs, but burned-out diodes give 1 outputs.

A very similar procedure is used in PROMs, except that each diode has in series with it a thin fusible link, often made of nichrome, which is blown rather than the diode itself. The diode is there strictly to prevent sneak current paths, which might steer the programming current through a different path and blow out more than one bit.

Experiment #58
Completing the ROM

Problem: That's all very nice, but how do you use a ROM like this on the address and data buses of a computer?

Solution: You can't connect it directly, because so far you have only wired the matrix. What we need next is some decoding circuitry for the input lines and Tri-state buffers for the outputs.

Procedure: Fig. 2 shows the complete circuit of a practical ROM. First of all, we can use a 7442 decoder as an address decoder. Since we have only four locations, we need only two address lines. We can use the A and B inputs to the decoder for address bits A0 and A1; this leaves two more inputs to the decoder, both of which must be low to permit the IC to ground one of the four outputs we are using. Hence these two inputs can be treated as active-low chip enables and labeled $\overline{CE1}$ and $\overline{CE2}$. The overbars imply that both inputs must be low for the circuit to be operational.

We put pull-up resistors on each of the output lines and connect them to a three-state output buffer such as an 8097, 8T97 or 74367. In order to enable the buffer, we have an output enable line labeled $\overline{OE}$; here again, the overbar implies that this line must go to ground for the outputs to be enabled.

To observe the outputs of the buffer, connect your diodes and resistors as before. Now ground the two chip enable inputs and the output enable; then put various combinations of 0 and 1 on the two address inputs A0 and A1, and watch the outputs change as you address different locations. Each burned-out diode should provide a 1 output, while the remaining diodes should provide a 0 output.

Theory: The circuit of Fig. 2 is fairly straightforward, with the 7442 grounding an input line selected by the two-bit address, and the 8097 outputting the selected four bits.

The existence of so many chip enables and output enables is a little confusing, though. Actually, when you think about it, the two chip enables to the 7442 are not needed; we might as well ground them permanently. If the output isn't enabled, it doesn't matter what the rest of the ROM does. We can rely on the $\overline{OE}$ to do the chip selecting.

In this way, the ROM memory will decode all addresses and provide data to the output buffers for all memory operations, even those not intended for this particular memory. But the output buffers will provide an output only when this one ROM is

Fig. 3. An impractical way of building a 16 x 1 memory.

selected.

This is really the best way of going anyway. One of the things to worry about in any fast system is glitches. A glitch is a very short, unwanted pulse on some line, present either because of noise or because not all parts work at the same speed. If we were to combine the chip select circuit with the addressing circuit (by using the extra inputs to the address decoder for enabling), then we would always be selecting the memory at the same time as we would be providing it with an address. There would be a short time when the memory is already enabled but not yet settled down to the correct address. Its output would be nonsense briefly, until it had a chance to stabilize. The best procedure is to give the memory an address, get it started looking for the contents of that address, and then, just before we want the data, give it the enable and be reasonably sure that the output has settled down to the correct value.

There are some memory ICs that do provide separate chip enables and output enables. In some, this is because there may be latches in either the address decoders or in the output buffers; in others this may be done to conserve power by only operating the memory matrix when the IC is being used.

Experiment #59
Building a 16 x 1 ROM

Problem: The ROM we just built had four locations with four bits each; that is, it was a 4 x 4 memory. What does the matrix look like if we need a 16 x 1 memory instead?

Solution: One solution might be to use a long, stringy matrix like that of Fig. 3, but this is not a good approach. In this case it isn't too bad, but can you imagine what a 4096 x 1 memory would look like?

This is a bad approach for several reasons. First of all, it is not an efficient use of semiconductor real estate. If you built the IC chip like this, it would be so long and narrow that it would tend to break. If you tried to fold the design into a Z shape, then the connections on the IC chip would become very messy. Finally, the address decoder would be a horror.

A better approach is to keep the matrix as close to a square as possible and add a multiplexer at the output.

Procedure: In Experiment #44 you used a 74150 multiplexer; now it's time to dust it off and put it back to work in the circuit of Fig. 4. Keeping in mind that the 74150 inverts its output so that a burned-out or missing diode generates a 0, use four lengths of wire to connect various combinations of 0s and 1s to the four address inputs. Note that each different address provides a different output.

Theory: What we are doing here is splitting the four-bit address needed to select one of the 16 locations into two parts. Two bits, A0 and A1, select one of the four vertical input wires; the other two, A2 and A3, select one of the four horizontal output wires and feed its output to the 74150 output on pin 10.

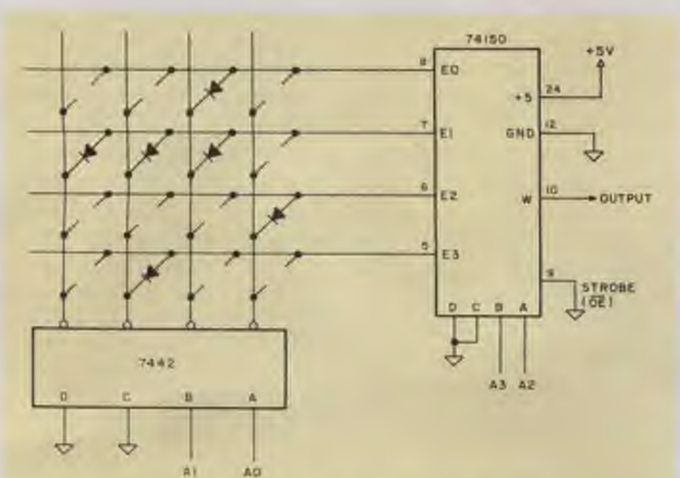


Fig. 4. A 16 x 1 ROM using a square matrix.

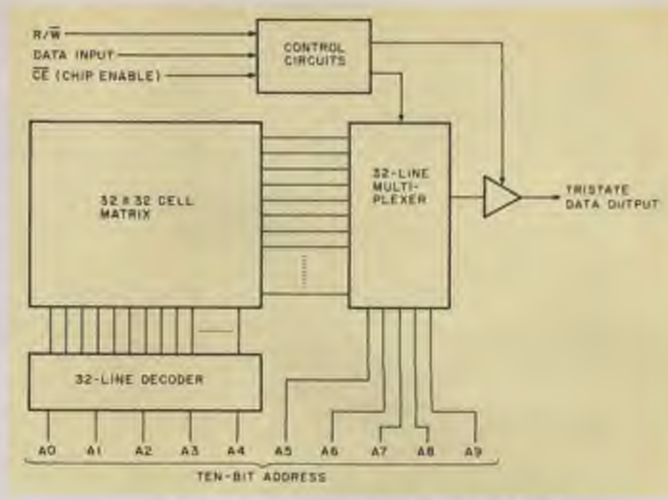


Fig. 5. Block diagram of 2102 1024 x 1 RAM.

The 7442 could select up to eight input wires, and the 74150 up to 16 outputs; so with these two ICs we could actually expand the matrix to an 8 x 16, for a total of 128 bits. It's obviously not a practical thing to do on the console breadboard, but it could be done. The 74150 even has a strobe input on pin 9, which could be used as an output enable.

Now we are on the right track—by building the matrix in a square shape, we can provide many bits of memory with relatively few vertical and horizontal interconnections. For instance, a 4096 x 1 memory could be built with a 64 x 64 matrix, having a total of only 128 wires. Doubling the size of the matrix to 128 x 128 would give us four times the number of bits, a total of 16K.

This explains why almost all popular memory integrated circuits use a square matrix; it is by far the most economical. In fact, the construction of a static RAM is not much different from the diode ROM; there is still a matrix of memory elements, with some addressing circuitry to select a vertical wire or column, and a multiplexer to select a horizontal wire or row. The only difference is in the type of circuit used to store one bit and in the precise way it is addressed.

The actual storage element for one bit is called a cell. In the diode ROM, the cell was a single diode; in an IC ROM, the cell would be a diode in series with a fusible link; in a static RAM memory, the cell is a flip-flop; in a dynamic RAM the cell is a capacitor.

Fig. 5 shows the block diagram of a popular memory IC, the 2102 1K x 1 static RAM. This IC in all its various forms has been the workhorse of small computers for the past several years, and is only just now being replaced by larger ICs.

A matrix of 32 x 32 cells stores 1024 bits. A 32-line decoder at the bottom and a 32-line multiplexer at the right select one row and one column of the matrix. Since this is a RAM, it has both a data input and a data output—in this case on two separate pins (some memory ICs use the same pin for both input and output to a bidirectional bus.) Two additional inputs are a R/$\overline{W}$ or Read/not Write control pin and a $\overline{CE}$ chip enable that enables both writing and reading. The output is a three-state output buffer.

MOS Handling Procedures

Before going on with the next experiment, let's talk about the correct way of handling MOS ICs.

MOS stands for metal oxide semiconductor; it essentially applies to any device that uses insulated gate field-effect transistors. In such a transistor, we have a silicon device that has a

ntrol electrode called a gate; e gate may be silicon, but ost often is metal. The gate is en separated from the silicon dy of the transistor by a thin yer of silicon oxide, which is insulator. Hence the name: etal gate, Oxide insulator, licon transistor.

The oxide layer is extremely in—just a few layers of oms. If the voltage between e gate and the rest of the transtor exceeds even a moderate vel, the insulator will break wn, and a tiny spark will jump rough it, punching a hole. nce that happens, the transisr is useless. Since the insular layer is thin, not much ltage is required to damage Furthermore, not much current is needed either.

As it happens, each input to MOS IC is connected directly the gate of a MOS transistor. hus it is easy to ruin an entire by exceeding the voltage mit on an input pin.

In particular, even static ectricity is enough to damage e IC. If you rub your feet on a rpet you may generate thouands of volts and get a real hock as you touch a light witch. But just walking norally, rubbing your hands on our polyester slacks or sitting own in a plastic chair may enerate enough voltage to zap n IC without your even feeling slight tingle. In other words, st touching a MOS IC may amage it without your even nowing about it.

IC manufacturers try to minize damage by including a rotection circuit on each input. This consists of a resistoriode network whose purpose to limit the applied voltage to safe level. But since static lectricity zaps are extremely apid, the IC may be damaged ven before the diode has a hance to conduct and liminate the danger. Hence, hese protection circuits help ome, but are not 100 percent ffective.

So the only safe way to prevent damage to such an IC is ot to touch it. Since that's not ossible, the next best precauon to take is, "Never let a tatic electricity voltage build up between you and the IC."

Since static electricity always involves small currents (at least, most forms do unless you talk about lightning), even a very high resistance between two objects is enough to prevent a voltage from developing. So you can never develop a dangerous voltage between two things already connected. For instance, if a MOS IC is connected to a circuit and you rest your bare elbow on the box the circuit is mounted in (assuming that the circuit is in some way grounded to the box), then there cannot be a voltage between your hand and the IC. If

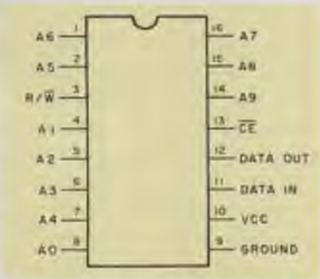


Fig. 6. 2102 RAM pin-out.

at this point you touch the IC pins, it is safe. (If, though, the box is connected to one side of the power line and you simultaneously touch a radiator, then the IC might be safe while you get fried.)

The point is that we have to apply a little common sense to MOS IC handling. Perhaps the best way to explain it is to describe some of the procedures I follow in my own lab.

When installing a MOS IC, I pick it up by the foam carrier it comes packed in before touching the pins themselves. When removing it from its carrier, I hold the carrier in one hand and the IC in the other and pull them apart. Then I place my fingers over as many of the IC pins as I can comfortably hold. Before installing it in a device, I reach for a convenient ground point on the device with my other hand and hold on while installing the IC. Obviously, I do this with the power turned off and the line cord unplugged.

If working on a printed circuit board, I raid my wife's kitchen for a two-foot piece of alumi-

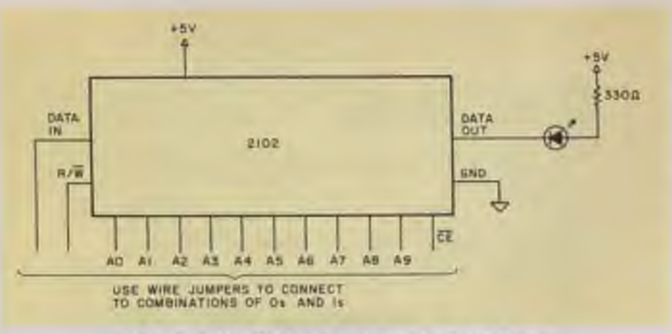


Fig. 7. 2102 RAM Experiment #60 wiring.

num foil, which I then spread out on my workbench. I keep the board on the foil and keep the ICs on it, too. I also try to keep my tools on it, as well as my elbow; I wear a short-sleeved shirt, so this makes a good connection. If my workbench had a metal top, then this precaution would not be necessary.

I use a soldering iron with a three-wire cord, so that the tip is grounded; but before using it I make sure to touch the tip to the foil worktop just in case. Some people suggest grounding the worktop, but this always makes me a bit worried, since it is possible to get a severe shock if you touch a grounded object at the same time as a hot one. An alternative would be to ground the worktop through a high-value resistor of 100k to 1 megohm. I may also use a cordless rechargeable soldering iron, when convenient, making sure to touch the aluminum foil just before soldering.

The purpose of the aluminum foil in all of this is to constantly keep shorting everything together—equipment, tools, ICs and me. That way there is never a chance to develop any damaging voltages between things that might touch the MOS IC.

Now, how to protect the 2102 IC we are going to use next? Follow the simple, common-sense rules above. Start with a blank prototyping socket on your breadboard and put in the ground wire from a ground point on your console to what will eventually be the ground pin of the IC, pin 9. Watch out: Unlike many of the 16-pin ICs you have used so far, this one does not use pins 8 and 16 for ground and power. Then get another wire, connect one end to ground on the breadboard console and hold the other end in your hand at all times. Now take the IC out of the package, plug it into the socket and hook up all the other leads while still holding onto the ground wire with your other hand.

At any point, you can quit, leave the unit and come back later. Just make sure to grab the ground wire before touching any other part of the unit.

Experiment #60
The 2102 Static RAM Memory

Problem: How can we test a 2102 RAM to get a quick idea of whether it works or not?

Solution: We can hook it up in a simple circuit, store a few bits and see whether they stay stored.

Procedure: Fig. 6 shows the pin-out of the 2102. Make the ground, power and LED connections shown in Fig. 7. Then use short wire jumpers to make the remaining connections to the data input, read/write, address and chip enable pins.

Although these MOS ICs are TTL compatible, their input and output terminals do not behave like normal TTL inputs and outputs. For example, TTL inputs assume a high input level (or 1) if no connection is made to them. This was convenient in a number of prior experiments, where we could simply leave an input disconnected to save time. MOS ICs do not like that—they may assume either a high or a low voltage level, or some level between; they may possibly even oscillate. You must make all the jumper connections to either ground (0) or +5 volts (1).

Likewise, TTL outputs can provide fairly sizable output currents; "TTL-compatible" MOS outputs cannot. That is why we changed the resistor in series with the LED from the 150 Ohms used earlier to 330 Ohms. In fact, if your LED will provide enough light, it might be better to go to even higher resistances, such as 470 Ohms or more.

Writing or reading from a RAM requires a specific sequence of carefully timed steps. Each of the inputs must be applied for a required time in order for the IC to properly accept the information and provide the right output. Read the following theory to see how this is done.

Theory: The sequence of steps required to use the RAM is shown in the manufacturer's data in the form of switching time waveforms as shown in Fig. 8 for a write and Fig. 9 for a read.

To store a bit in a certain address, follow this procedure; the number of each of the following steps is shown circled in Fig. 8:

1. Start with the chip enable and read/not write lines both high or +5 volts.

2. Place an appropriate address on the ten address lines. As shown in Fig. 8, to the left of the 2 dotted line, we have shown the address line dotted as well. This means that it doesn't matter what is on these lines at that time. But from time 2, all the way to time 10, the address lines must all be stable at either 0 or 1. This is why there are two curves drawn for the address lines, one high and the other at ground. Some of the lines may be 0 and the others 1, but all must stay steady during the entire write cycle.

3. Now ground the chip enable line.

4. Next, place the data to be written on the data-in line. The data must again be stable until after writing is done, up until time 8.

5. Then switch the R/W̄ line low for a write.

6. Now keep all the lines steady for a while.

7. Switch the R/W̄ line back high to stop writing.

8. Leave the data applied for just a while longer; then you can disconnect it at time 8.

9. Disable the chip.

10. Any time after that, remove the address.

The total time for a write, from time 2 to 10, is called the write cycle time. For a standard 2102 this may be one microsecond, although there are faster versions.

A manufacturer's specification sheet will give various minimum times for the sequence shown in Fig. 8. For instance, the sheet may say that at least 200 ns must pass from the time that the address is applied to the time the read/write line goes low, or that the data must remain stable for at least 100 ns after the read/write line goes back high. In your experiment you do not have to worry about these minimum times since you will be working in seconds rather than billionths of seconds, but if you were designing a computer system you would have to read the spec sheets very carefully.

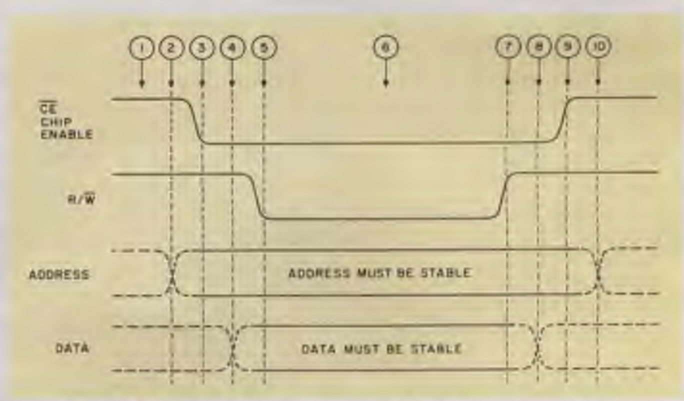


Fig. 8. Switching time waveforms for a 2102 write cycle.

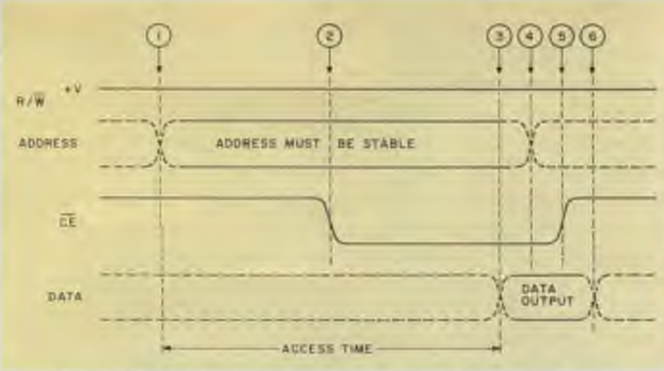


Fig. 9. Switching time waveforms for a 2102 read cycle.

Fig. 9 shows the switching times for a read cycle. To read, we would have to follow this sequence of steps:

1. Starting with the read/write and chip enables both high, apply an address.
2. Wait for the address decoders in the IC to work, and then make the chip enable low.
3. Wait for the data to come out.
4. and 5. Remove the address and chip enable. These could be removed together, or the address could be disconnected slightly earlier as shown. In your experiment, though, disable the IC first.
6. The data will still come out for a short time after the chip enable goes back high, but don't count on it.

The total time for the entire cycle is called the read cycle time, but another time is often quoted and that is the read access time shown in Fig. 9. This is the time from when the address is first applied to the time data first starts to come out. Typical access times for small home computers range from 600 ns down to as low as 250 ns. Much faster RAM memories are available, but not yet used to any great extent in small computers since they are faster than the microprocessors used with them. Very fast memory ICs would be used in large computers that use TTL or ECL ICs in the processor.

Comment: Although this experiment does a quick check of how a 2102 memory operates, it is really not a thorough check. To do a complete test, we would have to store a variety of bits into different locations, make sure each bit gets read out correctly and also check that writing into one location does not affect any other location. In general, it is a fairly lengthy test and, although we could design a special purpose tester to test 2102 chips, it would be easier to just do a quick check, install it in the computer and then write a computer program that would exhaustively test memory under program control.

More Theory: Static and Dynamic RAMs

It's not practical to do an experiment with dynamic memories, so let's just talk about them.

As mentioned earlier, the major difference between different memories is the type of memory cell used in their memory matrix. We mentioned that a static RAM uses a flip-flop as the basic cell, whereas a dynamic RAM uses a capacitor. Fig. 10 shows the circuits of two typical cells.

The static cell has five gates, using at least five transistors. There is one vertical input line, but two horizontal lines for each row of cells; one of these is used to write in a 0 and the other to write a 1.

Gates 3 and 4 make up the flip-flop that acts as the main memory element. Gates 1 and 2 are used to gate 0 and 1 inputs into the flip-flop, while gate 5 lets the output get from the cell back to the horizontal output wire.

Now compare this with the cell for the dynamic flip-flop. Here the cell consists of just

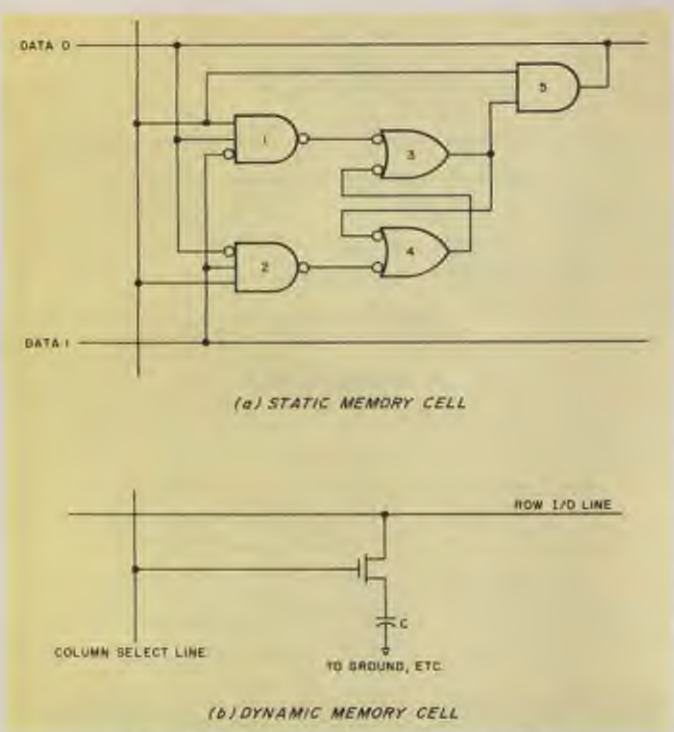


Fig. 10. Two types of RAM memory cells.

one transistor and a capacitor, with the capacitor actually a part of the IC structure and not a separate component at all. Seeing how much simpler the dynamic cell is, we can understand why dynamic memory ICs are cheaper and smaller.

In normal operation, the transistor in the dynamic cell is off or nonconducting, and the capacitor is disconnected. If it were a perfect capacitor, then whatever voltage it is charged to would stay there forever. In practice, though, there is some leakage, so the capacitor will stay charged for only a little more than two milliseconds.

Whenever a vertical input line—called a column select line—is turned on, the transistor turns on and connects the capacitor to the horizontal or row input/output line.

For a write to memory, the capacitor can be charged or discharged by placing either a high or low voltage on the row I/O line and then turning on the column select line. Once this is done, the capacitor will remember the bit for about two milliseconds.

For a readout of a bit, the column select line is again turned on. The voltage on the capacitor is connected to the row I/O line and detected by a sense amplifier. If the voltage is low, the sense amplifier will detect this as a 0 and force the voltage even lower. If the voltage is high, then the sense amplifier will detect this as a 1 and force the voltage even higher. This has the effect of reading the bit out of memory and also recharging the capacitor back to its original voltage. As long as a read is done at least once every two milliseconds, the capacitor will keep being recharged and will continue to hold the original bit. This is called refreshing.

In practice, each row I/O line has its own sense amplifier. So whenever any column select line is turned on, all the cells on that column get refreshed at the same time. Thus, it is only necessary to cycle through all the columns once every two milliseconds. This is what is done by the refreshing circuits on dynamic memory boards.

Since dynamic memory ICs can use smaller chips than static ICs of the same capacity, it makes sense to use smaller IC packages as well. Thus, there are many fairly large dynamic chips that use only 16- or 18-pin packages. To save on pins, many of these use multiplexing on the address pins. For example, Fig. 11 shows the block diagram of an Intel 2117 16K dynamic RAM. Although 14 bits are needed just to address 16K bits, this IC needs a total of only 16 pins altogether. This is done by breaking up the 14-bit address into two 7-bit chunks. One group of seven bits is called the column address and selects one of 128 columns in the matrix; the other seven bits are the row address and select one of 128 rows. The two 7-bit chunks are sent into the IC separately, one after the other, over the same seven address pins.

Instead of one large 128 x 128 matrix, this IC has two 64 x 128 arrays (array is another word for matrix) with the sense amplifiers connected in the middle between them.

There are several new signals shown here: $\overline{RAS}$ and $\overline{CAS}$ are row address strobe and column address strobe signals, which tell the IC which of the two address halves is on the address inputs. $\overline{WE}$ is a write enable.

The second advantage of breaking up the address bits as shown here is that the entire memory can be refreshed once every two milliseconds simply by sending in a whole batch of different column addresses.

Cycling through all the 128 column addresses can be done either by circuits on the memory board or by the processor itself. For instance, the Z-80 has the capability of refreshing dynamic memory automatically.

Preview

We have now finished looking at memory circuits and ICs. Before we can go on to look at a complete computer design, we now have several more things to look at: various ways of connecting input and output devices, the control signals used by microprocessors and how to control them, and the internal organization of a microprocessor.

Next time we will tackle the input/output problem. Stay tuned! ■

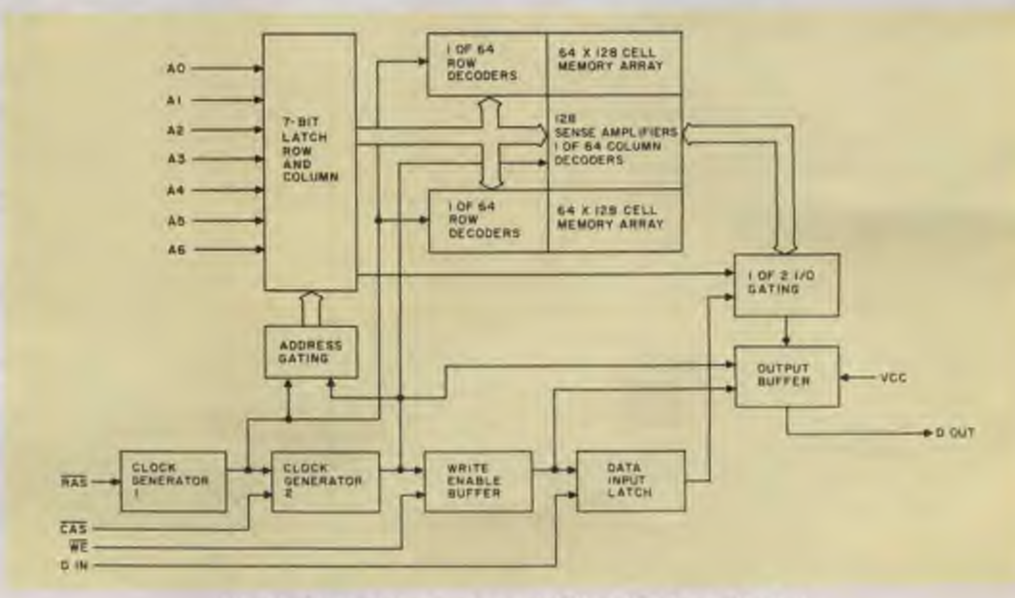


Fig. 11. Block diagram of an Intel 2117 16K dynamic RAM.

We've gathered the family to show you why

PERCOM's™ Number 1

in cassette data systems for microcomputers.

Pardon us for doing a little boasting, but we're proud of our family. Proud of each member's reputation for performance and reliability. And pleased that we can offer the best in cassette data systems and data terminal interfacing at low, home-computing prices.

It took more than guts and a little luck to forge a position of leadership. We're number 1 because you get more when you buy PERCOM™. The reason, simply, is experience. Every product described in this ad is based on nearly 10 years of crucial involvement in the design and manufacture of computer peripherals that use cassettes for mass storage.

Experience. It's why we developed a more reliable data cassette for home computing. Why our interfacing units provide **both** cassette and data terminal interfacing. Why you get the fastest, most reliable cassette data rates from PERCOM™. Experience. It's *the* reason for PERCOM™.

For your SS-50 bus computer — the CIS-30+

- Interface to data terminal and *two* cassette recorders with a unit only 1/10 the size of SWTP's AC-30.
- Select 30, 60, or 120 bytes per second cassette interfacing, 300, 600 or 1200 baud data terminal interfacing.
- Optional mod kits make CIS-30+ work with *any* microcomputer. (For MITS 680b, ask for Tech Memo TM-CIS-30+—09.)
- KC-Standard/Bi-Phase-M (double frequency) cassette data encoding. Dependable self-clocking operation.
- Ordinary functions may be accomplished with 6800 Mikbug™ monitor.
- Prices: Kit, $79.95; Assembled, $99.95.

Prices include a comprehensive instruction manual. Also available: Test Cassette, Remote Control Kit (for program control of recorders), IC Socket Kit, MITS 680b mod documentation, Universal Adaptor Kit (converts CIS-30+ for use with any computer).

MIKBUG® Motorola, Inc.

For your S-100 computer — the CI-812

- Both cassette and data terminal interfacing on one S-100 bus PC board.
- Interfaces *two* recorders. Record and playback circuits are independent.
- Select 30, 60, 120, or 240 bytes per second cassette interfacing, 110 to 9600 baud data terminal interfacing.
- KC-Standard/Bi-Phase-M (double frequency) encoded cassette data. Dependable self-clocking operation.
- Optional firmware (2708 EPROM) Operating System available.
- Prices: kit, $99.95; assembled, $129.95.

Prices include a comprehensive instruction manual. In addition to the EPROM Operating System, a Test Cassette, Remote Control Kit (for program control of recorders), and an IC Socket Kit are also available.

For your data storage — Pilon-30™ data cassettes

- Orders-of-magnitude improvement in data integrity over ordinary audio cassettes.
- Pilon-coated pressure pad eliminates lint-producing felt pad of standard audio cassettes.
- Smooth pilon coating minimizes erratic tape motion.
- Foam pad spring is energy absorbing. Superior to leaf spring mounted pad which tends to oscillate and cause flutter.
- Five-screw case design virtually precludes deformation during assembly.
- Price: $2.49.

PERCOM™ products may be purchased from home computer dealers nationwide, or may be ordered direct from the factory.*

*Texas residents must include an additional 5% for factory orders. MC & Visa cards honored.

PERCOM™ *'peripherals for personal computing'*

PERCOM

PERCOM DATA COMPANY, INC.
DEPT. K
318 BARNES • GARLAND, TEXAS 75042
Phone: (214) 272-3421

P7

Motorola's Latest: The MC6802

The author really likes the 6800 microprocessor—but he likes the 6802 even better.

Tim Ahrens
7405 Ladybug St.
Austin TX 78744

By now, most of you have heard about the 6800 microprocessor and some of its capabilities. I think that it is the number one MPU device, but since all don't share my opinion, let's just say that it is one of the four most popular chips. Several different manufacturers have based their systems on the 6800, and software for it has been growing by leaps and bounds.

I'm not going to say much more about the 6800 itself, since much documentation has already been printed. Even though I am a strong 6800 supporter, I have found a device that surpasses it in many ways. It's a family member—the 6802, which combines all of the features of the 6800 plus the following:

1. On-chip clock circuit.
2. 128 x 8 bit on-chip RAM (0000-007F).
3. 32 retainable bytes of RAM (low-power standby).
4. Software compatibility with the 6800.

The on-chip clock circuit is just what it says; add an external crystal and capacitor to provide all the necessary clocks to the MPU (2-phase clocks, etc.). The specifications rate this crystal frequency at 4 MHz, but I have been running my circuit with a 5 MHz crystal.

Now, don't get your hopes up, Z-80 fiends; the CPU has a divide-by-four circuit in it to work with! Still, 1.25 MHz is nothing to sneeze at. When you consider the various problems of the "faster" CPUs, with the necessary wait states, etc., the extra speed doesn't really mean much—especially when used in the hobbyist mode.

The on-chip RAM is located from 0000 to 007F hex, with the first 32 bytes having the capability of being retained in a low-power mode by utilizing the Vcc standby, thus allowing memory retention during a power-down situation. This RAM can, of course, be turned off so it won't interfere with any external memory boards.

The 6802's Evolution

As you can see from the pinout diagram in Fig. 1, there are really only subtle differences in the two relatives; however, they are definitely not pin-for-pin compatible. Rather than repeat data of the entire device, and risk boring you, I will give only a brief description of the signals that are different than those on the 6800.

RAM Enable (RE). A TTL-compatible RAM enable input controls the on-chip RAM of the 6802. When placed high, the memory is enabled to respond to the 6802 controls; when low, the RAM is disabled. This pin may also be utilized to disable reading and writing the on-chip RAM during a power-down situation. The RE must be low 3 cycles before Vcc goes below 4.75 during power down.

EXtal and Xtal. These connections are for a series resonant fundamental crystal (at cut). A divide-by-four circuit has been added to the 6802 so that a 4 MHz crystal may be used in lieu of a 1 MHz crystal for a more cost effective system . . . maybe even try an inexpensive 3.579 MHz TV color burst crystal!

Enable (E). This pin supplies the clock for the MPU and the rest of the system. This is a single phase, TTL-compatible clock, and may be conditioned by a memory ready signal. It is equivalent to the $\Phi2$ on the 6800.

Vcc Standby. This pin supplies the dc voltage to the first 32 bytes of RAM as well as the RAM enable (RE) control logic. Thus, retention of data in this portion of the RAM of a power

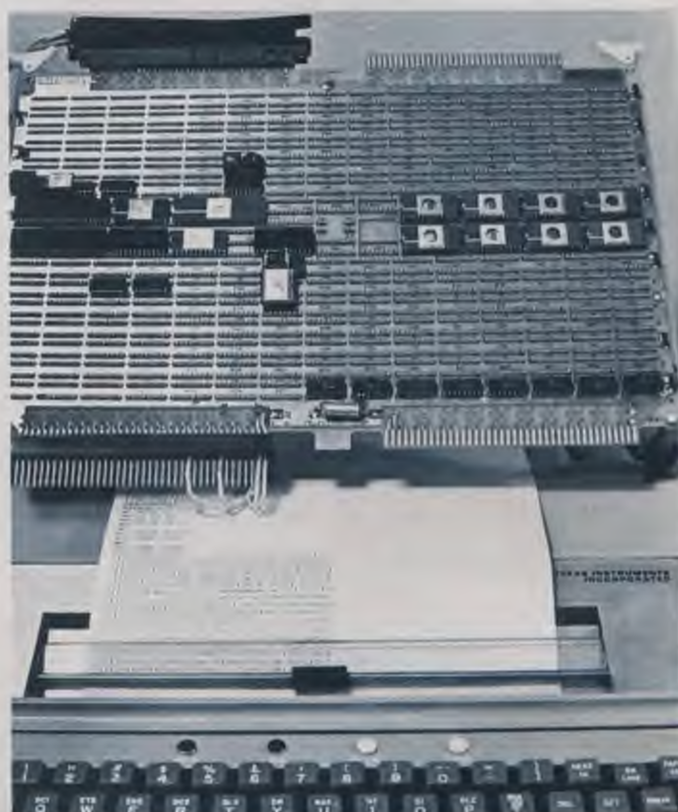


Photo 1. The basic system, shown here on a wire-wrap card, is the beginning of a portable computer system in a typewriter case with keyboard and MITE printer. To demonstrate the processor's capabilities, the system was connected to an RS-232 printer (a TI Silent 700).

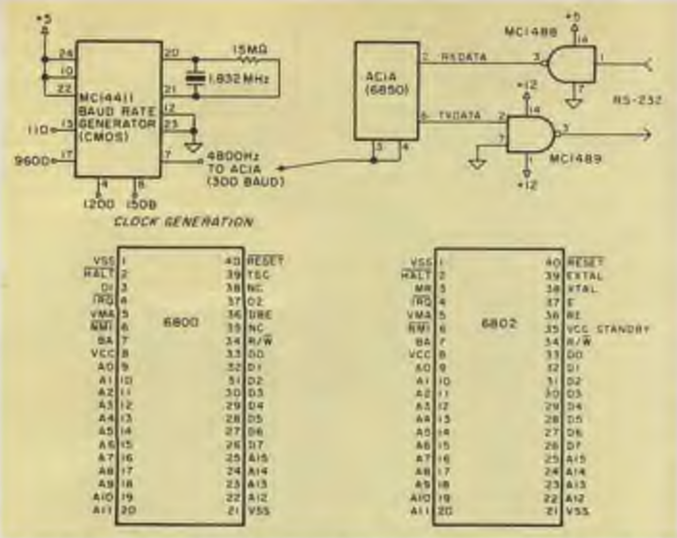


Fig. 1. 6800 vs 6802 pin-outs, RS-232 transmitter/receiver and MC 14411 baud rate generator. The 6800 and 6802, although not pin-for-pin compatible, are 100 percent software interchangeable.

up, power down or standby mode is guaranteed.

OK, so now we have a really nice processor. What can it do? First, a little history. . . .

Since the advent of the MC6800, the easiest monitor ROM to use was MIKBUG. It was taken by SWTP, and the first Motorola Evaluation Board was supplied with it. Designated the MC6830L7 ROM, it was designed to talk to a peripheral interface adapter (PIA), the MC6820. Although it worked quite well with this device, it was a bit difficult to change clock speeds, and was rather inefficient. Why use a parallel port for serial data?

Motorola soon realized these problems and developed the Minibug II and III. With these, the asynchronous communications interface adapter (ACIA) was utilized, and some other software development features were implemented.

Enter the 6846

The 6846, in conjunction with the MC6802, or even the 6800, makes a very cost effective microprocessor system. Here's why: The 6846 contains ROM, I/O and a timer all in the same package. It contains 2048 bytes of mask-programmable ROM, an 8-bit bidirectional data port with control lines, and a 16-bit programmable timer-counter.

At the present time, the ROM contains MIKBUG 2.0 with a high-speed audio cassette interface program, and is fantastic (more on that later).

General Description

Programmed storage. The mask-programmed ROM section is similar to other ROM products of the 6800 family. It is organized in a 2048 x 8 bit array to provide read only storage for a minimum microcomputer system. Two mask-programmable chip selects are available for user definition.

Address inputs A0-A10 allow any of the 2048 bytes of ROM to be uniquely addressed. Internal registers associated with the I/O functions may be selected with A0, A1 and A2, and the mask-programmed chip selects. The bidirectional data lines (D0-D7) allow the transfer of data between the MPU and the 6846.

Timer-counter functions. Under software control, this 16-bit binary counter may be programmed to count events, measure frequencies and time intervals, or similar tasks. It may also be used for square-wave generation, single pulses of control duration and gated delayed signals. Interrupts may be generated from a number of conditions selectable by software programming.

The timer-counter control register allows control of the interrupt enables, output enables and selection of an internal or external clock source. Input pin $\overline{CTC}$ (counter-timer clock) will accept an asynchronous pulse to be used as a clock to decrement the internal register for the counter timer.

If the divide-by-eight prescaler is used, the maximum clock rate can be four times the master clock frequency with an absolute maximum of 4 MHz. The gate input $\overline{CTG}$ accepts an asynchronous TTL-compatible signal that may be used as a trigger or gating function to the counter-timer.

A counter timer output (CTO) is also available and is under the software control via selected bits in the timer-counter control register. This mode of operation is dependent on the counter-register, the gate input and the external clock.

Parallel I/O port. The parallel bidirectional I/O port has functional operational characteristics similar to the B port on the MC6820 PIA. This includes eight data lines and two handshake control signals.

The control and operation of these lines are completely software programmable. This interrupt input (CP1) will set the interrupt flags of the peripheral control register. The peripheral control (CP2) may be pro-

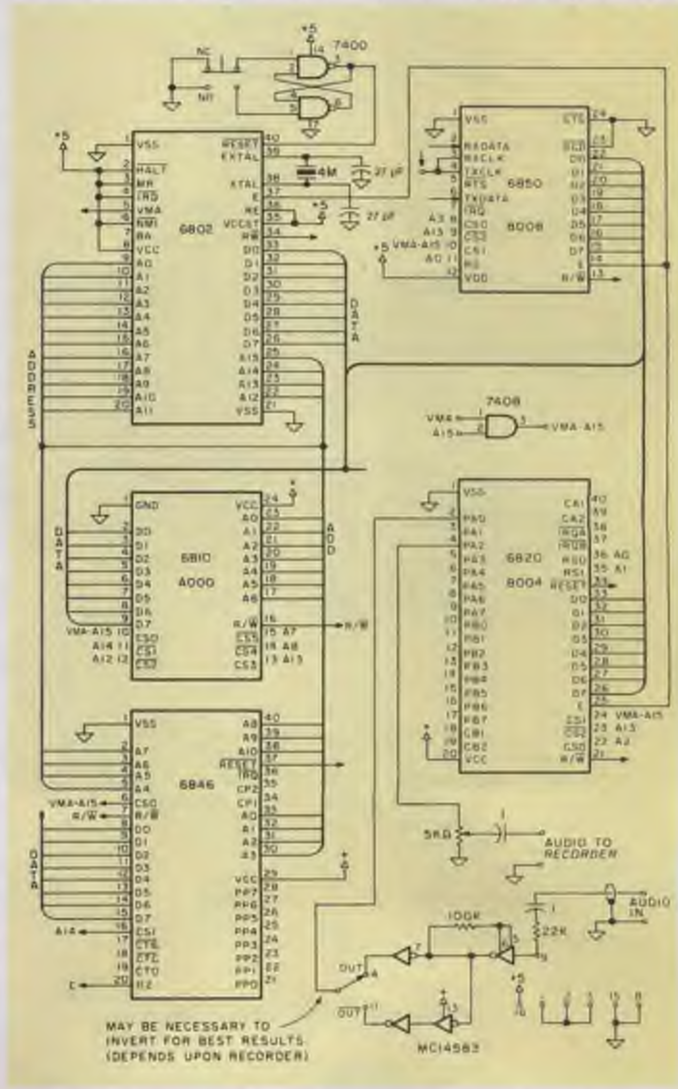


Fig. 2. This diagram is a very basic system layout using the 6802 and 6800 family of parts. It contains 256 bytes of RAM at 0000 (hex) and another 256 at A000 hex. This configuration can provide both serial and parallel data ports, as well as an audio cassette interface with file-searching capabilities. The system can be expanded (with appropriate buffers) to its full memory capability and 6800 software compatibility.

grammed to act as an interrupt input or as a peripheral control output. As an input or output, CP2 is compatible with standard TTL logic.

The only portion of the 6846 I will be using in this article is the ROM that contains MIKBUG 2.

The MIKBUG 2 is implemented in the MC6846 as a 2K byte firmware program for the development and debugging of 68XX/6846 systems. The ROM occupies the address space from F800 hex to FFFF hex. It is primarily intended to operate with the 6802, but will work quite well with even the plain vanilla 6800.

For MIKBUG 2 to execute, it assumes that there is RAM in address space A000 to A07F. It also assumes an MC6850 ACIA (serial port) at address 8008 hex. This ACIA interface allows MIKBUG 2 to operate with a Teletype or an RS-232 terminal, depending on the type of interface drivers provided.

An audio tape interface called Texortape is also provided in the ROM. This interface assumes a PIA at 8004 hex to implement this interface and uses one MC14583 for the receiver buffer from the audio cassette. It uses the A side of the PIA.

Table 1 is a summary of the features provided with MIKBUG 2.

The Texortape Cassette I
terface works great, and b
cause of the fast speed, co
pared to KC standard, for exa
ple, I have discarded the idea
using other types of fast me
ory access, such as a disk sy
tem . . . at least for a while!
will also search through voi
recordings for a file number.
over 3.5 million bits recover
at 2000 baud, I have had
failures using Memorex MR
and an inexpensive J.C. Penn
audio recorder (cat. # 851-001

The system that my 680
6846 is configured in will eve
tually be a small terminal with
MITE printer (see *Kilobau*
Nov. 1977, "Consider a MI
Printer," p. 38), with about 1
of memory, and 8K BASIC
EPROM. The board (see Pho
1) is of the wire-wrap variet
and is the only way to go.
that's connected are the bas
MPU, ROM, ACIA, PIA a
other external accessories (
2102 memory and the 8K BAS
EPROMS). The MC6802 a
6846 with MIKBUG 2 are ava
able at any Motorola Semic
ductor distributor.

My thanks go to Gary Po
for his photographic work.■

L	Loads formatted object taped into memory
M NNNN	Memory change at location NNNN
P	Print/punch ASCII formatted object tapes
R	Display contents of MPU user's registers
S1	Stop bit selection for ACIA (two stops)
S3	Stop bit selection for ACIA (one stop)
B	Print all breakpoints
C	Continue execution from current location
N	Execute next instruction
T NNNN	Trace NNNN instructions
G NNNN	Go to user program at location NNNN
D	Delete all breakpoints
U NNNN	Reset breakpoint at location NNNN
V NNNN	Set a breakpoint at location NNNN
E	Texortape Cassette Interface
	C—Check tape
	L—Load memory to tape
	D—Dump from memory to tape
	S—Set baud rate (400, 800, 1200, 1600, 2000 baud) (1200 baud data rate equals 1650 baud two-stop bit async!!!)

Table 1. MIKBUG 2 features.

Reference

MC6802 and MC6846 Advan
Information Data Sheets. Mo
rola, Inc., 1977.

TARBELL SETS STANDARDS
For Hobbyists and Systems Developers

Sales to thousands of hobbyists over the past two years have proven the Tarbell Cassette Interface to be a microcomputer industry standard. Tarbell Electronics continues research and development to produce new and efficient components to fill hobbyists' changing needs.

TARBELL CASSETTE INTERFACE

- Plugs directly into your IMSAI or ALTAIR*
- Fastest transfer rate: 187 (standard) to 540 bytes/second
- Extremely Reliable—Phase encoded (self-clocking)
- 4 Extra Status Lines, 4 Extra Control Lines
- 37-page manual included
- Device Code Selectable by DIP-switch
- Capable of Generating Kansas City tapes also
- No modification required on audio cassette recorder
- Complete kit $120, Assembled $175, Manual $4
- Full 6 month warranty on kit and assembled units

TARBELL FLOPPY DISC INTERFACE

- Plugs directly into your IMSAI or ALTAIR* and handles up to 4 standard single drives in daisy-chain.
- Operates at standard 250K bits per second on normal disc format capacity of 256K bytes.
- Works with modified CP/M* Operating System and BASIC-E Compiler.
- Hardware includes 4 extra IC slots, built-in phantom bootstrap and on-board crystal clock. Uses WD 1771 LSI Chip.
- Full 6-month warranty and extensive documentation.
- PRICE:
  Kit $190 ...... Assembled $265

CP/M with BASIC-E and manuals: $100

Compatible Disc Drives

Ask about our disc drives priced as low as $525.

TARBELL PROTOTYPE BOARD
Model 1010

- Gold plated edge pins
- Takes 33 14-pin ICs or
- Mix 40-pin, 18-pin, 16-pin and 14-pin ICs
- Location for 5 volt regulator
- Suitable for solder and wire wrap
- ALTAIR/IMSAI compatible

Price: $28.00

For fast, off the shelf delivery, all Tarbell Electronics products may be purchased from computer store dealers across the country. Or write Tarbell Electronics direct for complete information.

*ALTAIR is a trademark/tradename of MITS, Inc.
CP/M is a trademark/tradename of Digital Research

Tarbell Electronics
20620 South Leapwood Avenue, Suite P
Carson, California 90746
(213) 538-4251 T11

Ed Juge
2000 Thousand Oaks Dr.
Burleson TX 76028

TRS-80 Update: Level II BASIC

Here are the answers to questions asked most often about Radio Shack's new Level II BASIC.

From a user's point of view, the usefulness of any microcomputer is directly proportional to the "power" of the language it supports. Since many hobbyists are not professional programmers, BASIC will be their primary language. Although you might plan to learn machine or assembly language—and I say, "More power to you!"—the bulk of your programming will probably be in BASIC.

Radio Shack had you in mind when they built the TRS-80 with BASIC permanently stored in ROM, rather than read in from paper or magnetic tape. Your primary language is always there, yet you may program in machine or assembly language when you wish by CLOADing Radio Shack's T-BUG (monitor) or their Editor/Assembler from magnetic tape.

Radio Shack's Level I BASIC language is the ideal language to learn on, and is supported by an excellent tutorial user's manual. It will teach BASIC to anyone, even those with no prior experience. At some time, however, you'll find yourself ready to advance beyond the capabilities of Level I. At that point, Radio Shack's exchange of your Level I ROMs for Level II ROMs for only $99 is a tremendous microcomputer language buy. What does it buy you? Read on . . .

Level II is a 12K extended BASIC, as opposed to Level I's 4K, yet those who are using it will tell you it is much more than "three times the language." For starters, it increases the cassette baud rate from 250 to 500 baud. Identical programs store in less memory space, even though the numerous command abbreviations allowed in Level I are not supported in Level II. Execution times are significantly improved in Level II. Benchmark test no. 7 (see *Kilobaud* No. 10, p. 23) time improved from 110.1 seconds to 78.3 seconds in Level II.

Variables

Remember that one A(n) array allowed in Level I? Well, you can now have arrays named A through Z, A1 through Z9 and AA through ZZ. That's more than 900 arrays (limited, of course, by memory size)! In Level I you could have only one dimension, such as A(16), but you could not have a two-dimension array such as B(23,12). In Level II you can have multidimension arrays, of more than two dimensions . . . D5(23,4,12,5). Arrays may be designated as single-precision (six significant digits), double-precision (16 digits), integer (no decimal points) or string variable arrays (alphanumeric).

Yes, that also means that numeric variables can be double-precision to 16 significant digits. (Those of you making over 100 billion dollars per year can compute your income tax to the penny with a digit left over.) Integer variables from −32768 to 32767 are allowed. Numeric variables can also be designated as integer, single or double-precision.

In Level II you can now raise numbers to any desired power, for example, $143^{2.75}$, using the ↑ key. Trigonometric functions such as COS, ATN, TAN are included along with SQR, EXP, SIN and other more advanced mathematical functions not found in Level I. Numeric variable names allowed are the same as for arrays: A-Z, A1-Z9 and AA-ZZ. Integer, single-precision, double-precision and string variables are specified as A%, A!, A# and A$, respectively; and all of those just listed are distinct variables (i.e., they may exist in the same program at the same time, even though the variable names are all "A").

If you add them all up, Radio Shack Level II allows somewhere in the neighborhood of 3600-plus variables! But there's one more thing: Variables may have multi-letter names such as "PROFIT" or "LOSS." The only restriction is that you cannot have a "reserve word" (BASIC statement, command, etc.) nested inside one of your variable names, and you must remember that the TRS-80 recognizes only the first two letters of the name, so no two can start with the same first two letters.

String Variables

In Level I the TRS-80 could ask your name and mine, then call us by name while playing a game, but it could not compare our names. It could not tell which came first alphabetically, or even if your name and mine were the same. String variables were limited to 16 characters, and only two, A$ and B$, were allowed.

As indicated above, string variables under Level II may have the same names as other variables, and the length of each string may be up to 255 characters. (Remember those limits imposed by the amount of memory your TRS-80 has.) On "power-up," a total of 50 bytes of memory are allocated to strings. You may allocate more (or less) by the command CLEAR (n), with "n" being the number of bytes desired. This may be used as a program or command (keyboard) statement.

Full string handling—comparisons, concatenation, etc. —is allowed under Level II BASIC. Commands include: ASC(X$), which returns the ASCII code for the first letter of the string; FRE(X$), which returns the number of free bytes of memory allocated to strings; LEN(X$), which returns the number of characters in a string; LEFT$(A$,4), which returns the leftmost portion of A$, up to the fourth character; MID$, which returns a specified portion of the center of a string; RIGHT$, which returns the specified number of characters in the right side of the string; VAL(X$), which converts X$ to a numeric variable; STR$(x),

which converts the numeric variable X to a numeric string variable; and INKEY$, a neat little item that permits input of information while the program is executing (the program does not have to stop and wait for your input as with INPUT!). CHR$(x) returns the ACSII character function or graphic code for the number "X."

Cassette Operation

As stated, Level II raises the cassette baud rate from 250 to 500 baud. Level II cassette files are named; for example, you may specify CLOAD"STAR TREK." The recorder will bypass all files until it locates "STAR TREK," then that file will be loaded. After you've CSAVEd a program, you can rewind, type CLOAD?, and the TRS-80 will compare your SAVEd file with the one still in memory, byte for byte, verifying that it SAVEd properly. The SYSTEM command allows loading of machine-language cassette tapes.

Editing

Radio Shack's Level II BASIC allows you to edit program lines. Remember typing all of those lines over from scratch when a change within the line was required? No more! Simply type in EDIT n (n being the line number), and a dozen or so one-letter edit commands allow deletion, insertion or changing of a character or characters in the line without retyping the entire line . . . only that part to be changed. When a syntax error occurs, the offending line automatically enters the "edit" mode. Error codes in Level II, by the way, are much more comprehensive and include about 23 separate codes that are displayed when the error occurs.

Speaking of errors, you can "trap" them with Level II. An ON ERROR GOTO command sends the program to a specified line when any error occurs. The command is written into an early program line. When the subroutine is completed, a RETURN, RETURN NEXT (return to the line following the one where the error occurred) or RETURN n (return to line No. n) continues execution.

But, you say, "My program doesn't work; how do I know the error trapping will work?" Well, there is also a command ERROR n, which generates a specified type of error (n), to test it with. Your subroutine may include instructions to print ERR and ERL, which print the type of error, and the line in which it occurred, respectively.

DIM

What about those pesky "DIM" statements I find in programs in magazines and books? Well, DIMension simply sets aside memory space for arrays. It is used in Level II when the dimension of any array exceeds the number 10. In case your array is used in the first part of the program, and later on you need that memory but you're through with the array, simply insert a program line that says ERASE. That cancels the reserved space.

You might wish to permanently specify certain letters as specific names for a definite kind of variable, so you won't have to specify A! or K# each time you use them. DEFINT A,B,K-M specifies all variable names beginning with A, B and K through M as INTEGER variables. Likewise, you can use DEFSNG (single precision), DEFDBL (double precision) and DEFSTR (string variables). Variables can be converted from one type to another with CDBL, CINT and CSNG. All variables are set to "0" or "null " when CLEAR, RUN or NEW is executed.

Automatic Line Numbering

The command AUTO produces automatic line numbering. If that's all you type, it numbers lines beginning with line 10 and continuing in increments of ten. Typing AUTO 150 produces lines beginning with 150, in steps of ten, and AUTO 200,50 starts with line 200 and produces lines 200, 250, 300, 350, etc. Although Radio Shack didn't have enough space left in the 12K ROM to offer line renumbering, they are planning to offer that feature as a machine-language program on cassette. The USR(0) command allows access to machine-language subroutines.

Trace

How many times have you written a program that ended up in a never-ending loop, and you couldn't find out where it was hanging up . . . or one that seemed to be skipping a part of the program? Your only choice in Level I was to write in STOP commands at various places. Take heart! Level II allows you to tell your TRS-80, TRON (trace on).

As each line is executed, the number of that line appears on the video monitor, in brackets, so you can see exactly how the program is executing. When you've found the problem, type TROFF and return to normal operation. Between TRACE, editing, error codes and error-trapping routines, your debugging time should be reduced to almost nothing!

IF . . . THEN

Remember those extra lines when you wanted to say, "If something is true do this, but if not, then do that"? Well, now you can say "IF . . . THEN . . . ELSE . . ." and even "IF NOT . . . THEN . . . ELSE." Quite an improvement, wouldn't you say?

Formatted Print

The same PRINT AT (but now it's PRINT@) and PRINT TAB statements are used in Level II as in Level I. You can line up in-

formation on the screen with them, but remember those pesky decimal places that took a subroutine to "round off" so they'd look right? Well, with the command PRINT USING, and some format specifiers, you can round off to the desired number of decimal characters, insert commas to separate every three digits — instead of $25232.2367, you can have $25,232.24—specify a floating dollar sign, floating + or − signs preceding or following the numbers, and fill all number fields with preceding asterisks if you wish. Decimal points will line up, and you can deal in whole dollars or pennies, without fractions of a cent.

Other Features

By no means will this article cover all of the features of Radio Shack's Level II BASIC. The intention is to bring out those features which are not in Level I, and which hobbyists will find of greatest interest. All of the "standard" BASIC features found in Level I are included in Level II, except abbreviation of commands, and will not be dealt with again here. (By the way, I lied a little; two abbreviations are used in Level II: "?" for PRINT and "'" for REMarks.)

Level I-type TRS-80 graphics remain unchanged, except that they execute somewhat faster. Larger letters (32-character lines) are available by keyboard or program command in Level II, as is the nice "keyboard rollover" feature (you don't miss a letter if your finger hasn't released the previously typed key).

You advanced-programmer types will appreciate PEEK and POKE (allows looking at or placing a specified value in a specified RAM location); USR(0), which allows access to a machine-language subroutine (more USR calls will be allowed in Disk BASIC); and VARPTR(C), which gives you the address in memory where the variable "C" is stored. User-defined functions are not supported in Level II, although they should be in Radio Shack's Disk BASIC, which comes with their mini-disk drive. (Details of the DOS and Disk BASIC were not available at time of writing.)

Powering up a Radio Shack Level II TRS-80 immediately produces the question, "MEMORY SIZE?" The answer is *not* the number of RAM bytes in your machine. Unless you're using a machine-language subroutine, the answer is simply to press the ENTER key. The procedure for a machine-language subroutine is explained in the Level II manual, but the question is actually asking you to specify the upper limit of memory address available to BASIC. A complete memory map is included in the manual. Radio Shack's Level II manual is a user's manual rather than a teaching aid, as was Level I. It is well written and understandable, if you are already conversant with the Level I manual's contents.

Of course, Level II includes the necessary commands for input and output for printers, two cassettes, disk, etc., which are available with the use of Radio Shack's Expansion Interface. Although a number of disk commands are in the ROM, they are only the "hooks" that link Level II with Disk BASIC, and an attempt to use them will cause an error to occur.

In Level II program lines, logical lines and string variables all may be up to 255 characters long, although a maximum of 64 characters is recommended because of the limitation of the line length on video monitor. (It makes for easier reading.) Logical comparisons use the words AND and OR, rather than Level I's * and +.

It is likely that some of this information has excited some of you engineering types, so I'll answer your next question . . . even though it doesn't really have anything to do with Level II BASIC, the subject of this article. Radio Shack *will have* for sale later this year a TRS-80 service manual, including schematics and logic diagrams.

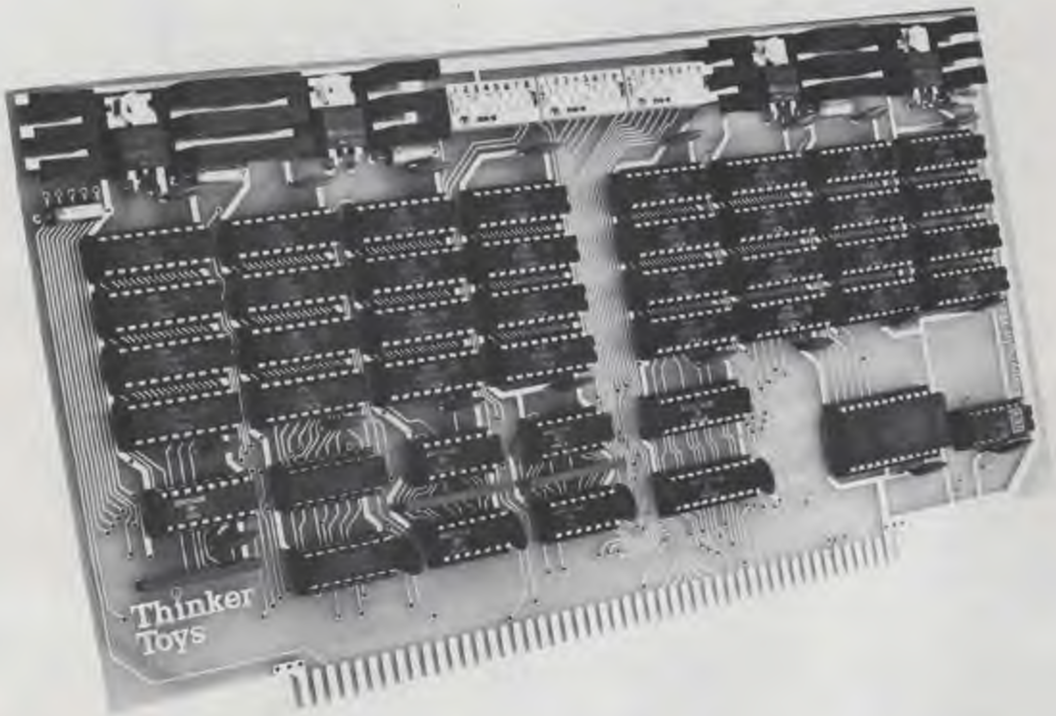
A list of statements, commands, etc., can be obtained by writing to: TRS-80 Computer Marketing, 1600 One Tandy Center, Fort Worth TX 76102. That list will include a brief description of all commands, more extensive than has been included here.

The bottom line is that Radio Shack's Level II BASIC seems to be a very powerful and very complete microcomputer language. I have used several excellent languages on both hobby and business microcomputers, and although your evaluation may vary depending on what you "grew up with," I think you'll find Level II to be one of the best micro languages on the market. If your primary interest is programming in BASIC, then having it in ROM is a definite plus.

One last point: Obviously, your Level I (250 baud) tapes won't run on a Level II machine. Radio Shack includes on cassette with each Level II kit an ingenious machine-language program that converts your Level I programs to Level II format. You load the conversion tape in the SYSTEM mode, load your Level I tape, press ENTER, then CSAVE the result on tape as a Level II program! What more could you ask for?■

SuperRam™

16K STATIC FOR $299



Introducing SuperRam™ 16K static memory, the one that's leaping tall price barriers at a single bound. It saves you about $100 on the usual cost of a big 16K memory for your S-100 system.

SuperRam™ 16K is the latest in cost-efficient memory designs by George Morrow, designer of the best-selling ECONORAM* memories.

SuperRam™ 16K is configured as four independent 4K blocks, each separately addressable and write-protectable. Designed to meet the proposed IEEE Standard for the S-100 bus (see IEEE Computer, 5/78), all signals are fully buffered—including address and data lines. And Morrow's design uses just 11 chips to keep the board uncrowded and trouble-free.

SuperRam™ 16K comes as an easily assembled kit, with solder mask and parts legend.

Ask for the SuperRam™ 16K memory kit at your local computer shop. Or if unavailable locally, call your BankAmericard/Visa or Master Charge order to 415-547-7548, 10-4 Pacific Time. Or send check or money order to Thinker Toys™, 1201 10th St., Berkeley, CA 94710. Add $3 for handling; Cal. res. add tax.

*ECONORAM is a trademark of Godbout Electronics

A product of Morrow's Micro-Stuff for

Thinker Toys™

T28

1201 10th Street
Berkeley, CA 94710

James Grina
1284 Fifield Pl.
St. Paul MN 55108

Super Cheap 2708 Programmer

The advantages of programming your own 2708s are many. This easy to build programmer enables you to do the job—inexpensively.

As a KIM owner, I have spent many enjoyable hours running and programming my micro. But as I wrote more programs, I noticed that some of the subroutines were used over and over in different programs. I began looking for a suitable PROM on which to store them. The price and availability of the Intel 2708 PROM were attractive, so I looked for a programmer. Since there were no 2708 programmers compatible with KIM, I sat down, equipped with the *Intel Memory Design Handbook,* a prototyping board and some components, and made one.

Software

In theory, the 2708 PROM has a few simple requirements for entering data. A blank or erased 2708 has all bits in the high, or 1, state. Information is entered by selectively programming 0s into the desired locations. To enter data, the $\overline{CS}$/WE pin is raised to + 12 volts to put the chip into the programming mode, the address of the word is selected and the data word is presented to the data pins.

After the data has settled (10^{-6} sec.), a programming pulse of + 26 volts at 20 mA is applied to the programming pin. The next address is then selected and the process continued until all addresses have been programmed. This is defined as one programming loop.

To determine the number of programming loops required to ensure valid data entry, use the following formula: N x t_{PW} greater than 100 msec, where N is the number of programming loops, t_{PW} is the programming pulse width and can be varied from 0.1 to 1.0 msec. For rapid programming, t_{PW} should approach one msec. This then requires 100 programming loops; the time required for complete

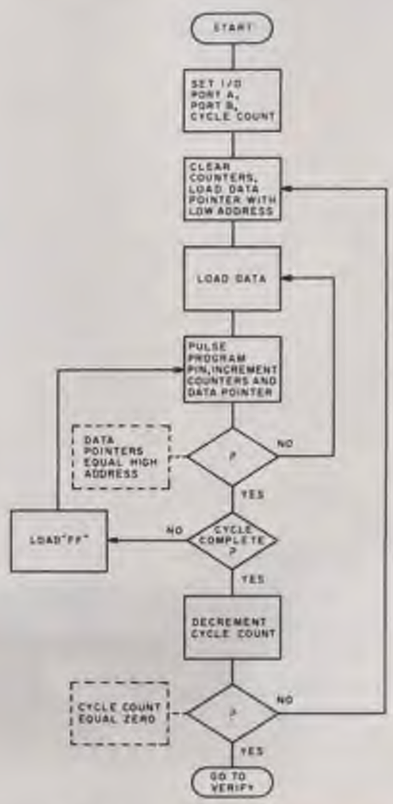


Fig. 1. Flowchart of main program.

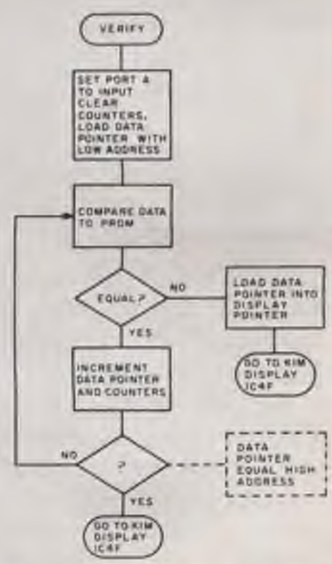


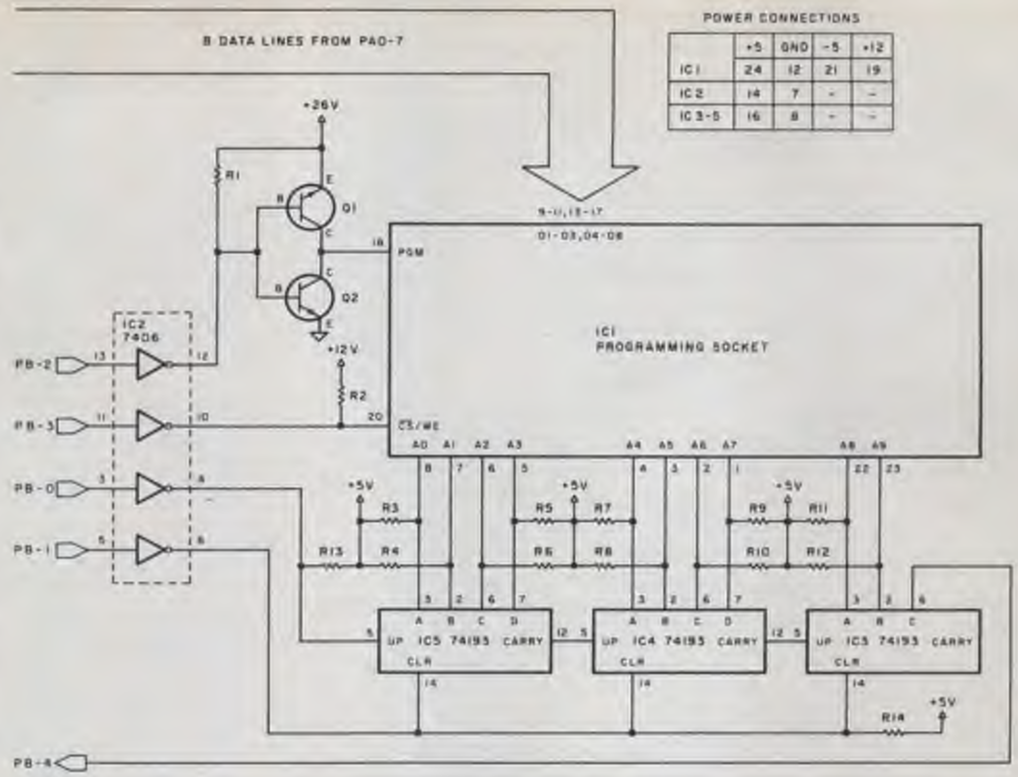
Fig. 2. Flowchart of verify program.

programming of 1K bytes is 1024 addresses times 1.0 msec times 100 programming loops, or about 103 seconds.

For KIM to perform the programming, it must control ten address lines, eight data lines, $\overline{CS}$/WE and programming pins. To do this directly KIM would have to be able to latch 20 I/O lines, but since the addresses are stepped through sequentially and completely, counters can be used to provide the address. Then KIM can use two lines to control the counters and a third to sense the end of the programming loop.

Since KIM has 15 I/O lines, this is perfectly adequate. The counters selected were 74193s because their pin-out gives a simple PC board layout; but any binary counter could be used.

The I/O port is connected to the up-count, the clear and the output of the 11th counting stage. The program pulses the clear, then toggles the up-



	+5	GND	-5	+12
IC1	24	12	21	19
IC2	14	7	-	-
IC3-5	16	8	-	-

Fig. 3. Schematic diagram of programmer.

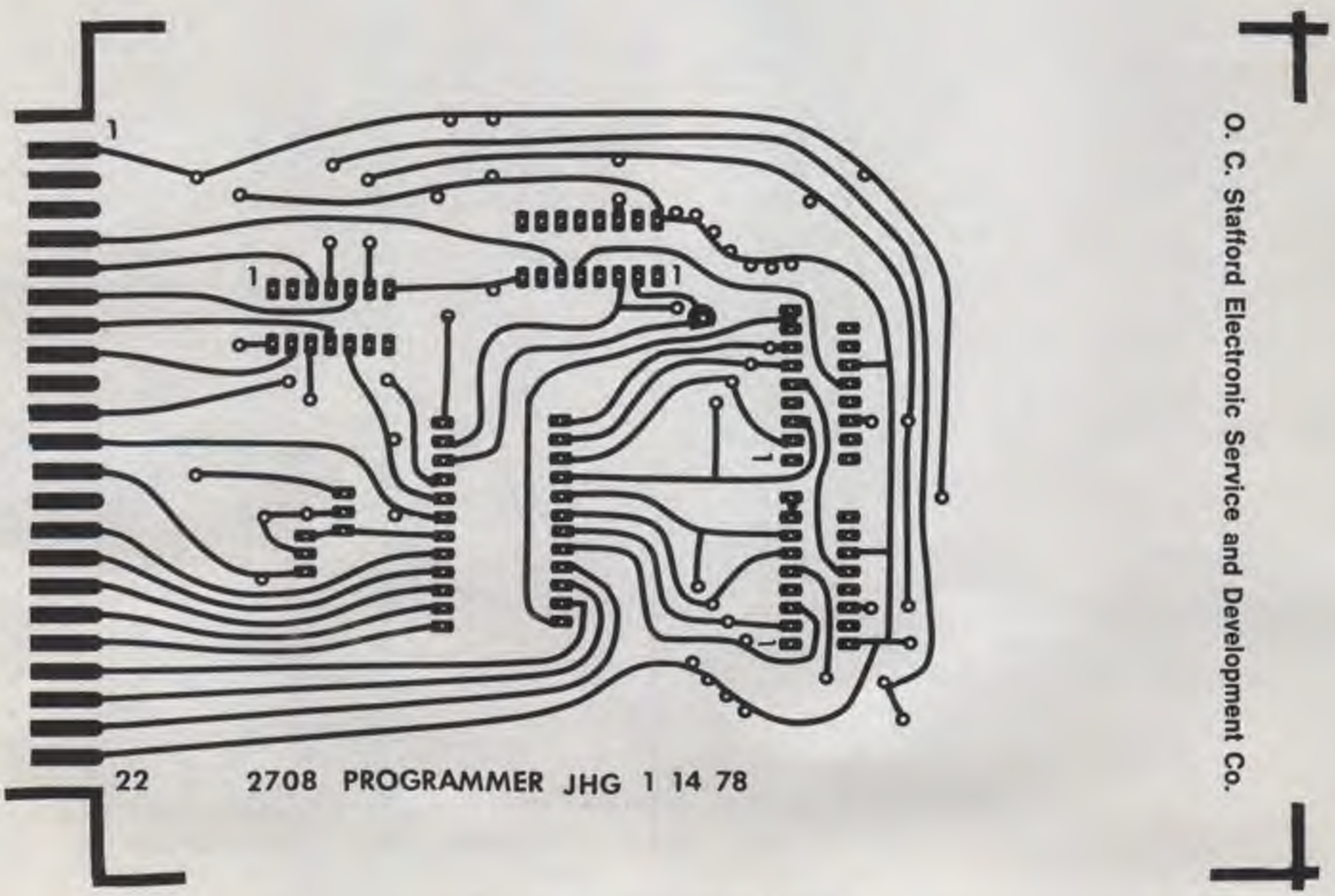


Fig. 4. Foil pattern for the PC board.

Model	Power Rating	Time of Exposure
S-68	12000uW/cm²	10 min.
S-52	12000uW/cm²	10 min.
UVS-54	5700uW/cm²	30 min.
R-52	13000uW/cm²	10 min.
UVS-11	5500uW/cm²	30 min.

All exposures are at a distance of one inch with the filter removed.

Table 1. Exposure guide for Ultra-Violet Products lamp. (Ultra-Violet Products, Inc., 5114 Walnut Grove Ave., San Gabriel CA.)

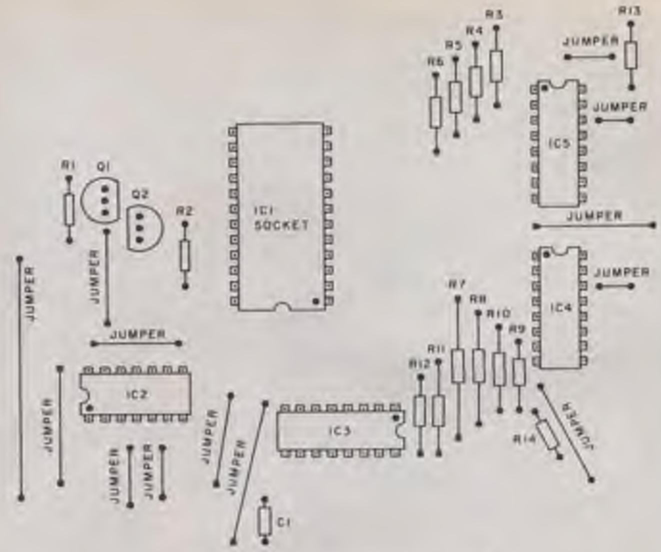


Fig. 5. Component placement.

IC1	2708 socket
IC2	7406 hex inverter (O.C.)
IC3-IC5	74193 binary counter
R1-R14	4.7k Ohm ¼W resistor
C1	0.1 uF capacitor
Q1	2N222 or 2N3904
Q2	2N3906

Table 3. Parts list.

count and tests the 11th stage output. When the end of the programming loop is detected, the counters are cleared and the cycle register, 00EE, is decremented (Fig. 1 is a flowchart of this main program).

Programmer	
1	GND and KIM A-1
2	N.C.
3	N.C.
4	KIM A-13
5	KIM A-9
6	KIM A-10
7	KIM A-12
8	KIM A-11
9	N.C.
10	-5 volts
11	+12 volts
12	+26 volts
13	N.C.
14	KIM A-8
15	KIM A-7
16	KIM A-6
17	KIM A-5
18	KIM A-2
19	KIM A-3
20	KIM A-4
21	KIM A-14
22	+5 volts

Table 2. Connector assignments.

To control the $\overline{CS}$/WE and programming pins, the lines are buffered with a 7406 (hex inverter with 30 volt open collectors). The programming pin requires a source of 26 volts at 20 mA and a sink of 1 to 2 mA, so push-pull transistors were used to control the actual input to the programming pin.

The program (listed in Program A) uses 00E7 through 00EE as address and counting registers on page zero and it occupies an additional 184 bytes of memory—either RAM or ROM. As written, it occupies page zero, which permits any RAM over 0200 as the source of data. It exits to the KIM monitor at 1C4F.

The body of the program consists of: an initialization routine that sets up the I/O ports, data pointer and cycle count; presentation of the data word and programming pulse; incrementing the counters and data pointer; testing for last data entry; testing for complete programming loop; and testing for end of programming.

When programming is completed, the routine jumps to a verify loop that compares the PROM contents to the data (see flowchart in Fig. 2). When the number of data words is less than 1024, the program enters FF into the remaining addresses. This permits additional entries to be made at a later date.

During verification, if an error is detected, the program jumps to the KIM monitor and displays the address of the error. Complete programming takes just under two minutes. Additional data can be entered into addresses containing FF, but all previous data must also be entered. All addresses must be programmed during each programming cycle.

The 2708 PROM can be erased by exposure to ultraviolet light at a wavelength of 2537 Å. The recommended dosage is 10 Watt-seconds per square centimeter. This can be determined by a UV meter or by following the exposures given in Table 1.

Hardware

A schematic of the programmer is shown in Fig. 3. The layout shown in Figs. 4 and 5 is a minimum hardware layout using a single-sided PC board

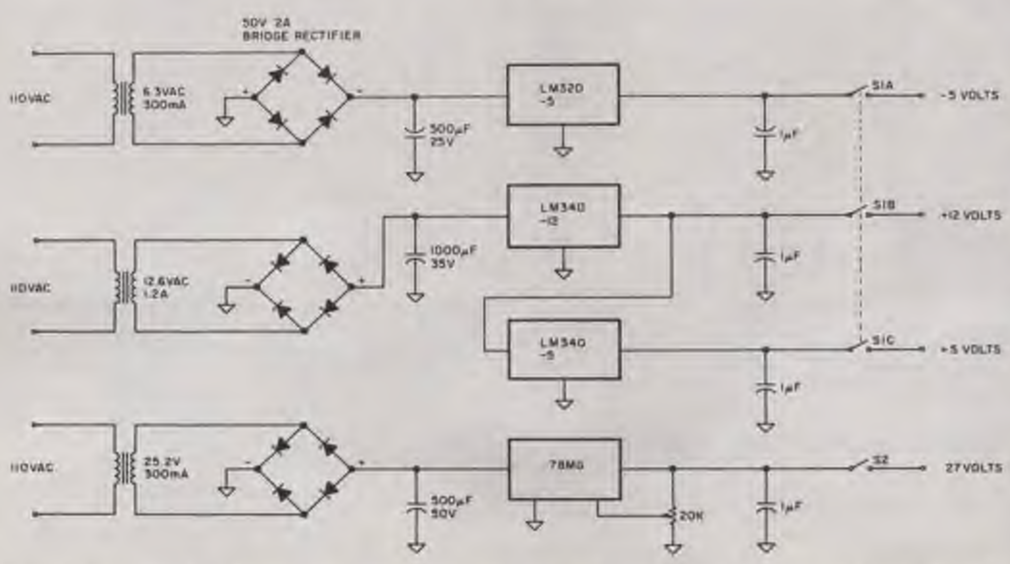


Fig. 6. Power-supply diagram. Adjust the output of the 78MG to give 26 volts after the drop across Q1.

with jumpers. It connects to the application connector supplied with KIM and the power supply. KIM's peripheral port A supplies the data to be entered; port B is used to control the counters and programming lines.

PB-0 is used to increment the counters, while PB-1 clears them. PB-2 controls the programming pulse and PB-3 controls $\overline{CS}$/WE. PB-4 is used to sense the completion of the programming loop by detecting a high on the C output of the high-order 74193. PB-5 and PB-7 are not used. The interface connections are shown in Table 2. Pull-up resistors are used on the TTL outputs to ensure that the logic levels required by the 2708 inputs (3.0 volts, V_{IH}) are met.

The programmer requires four voltages, in addition to the 5 volt supply for KIM; the 2708 requires a -5 volt supply at 45 mA, +12 volts at 65 mA and +26 volts at 20 mA. The diagram in Fig. 6 illustrates a suitable supply. The power for the programmer should be controlled separately so the programmer can be turned off when the PROM is being inserted or removed.

Programming Procedure

1. Turn off all power. Then plug the programming board into the application connector on the KIM. Insert the PROM to be programmed into the 24-pin programming socket. Check the orientation to make sure that the number one pin is in the proper alignment.

2. Turn on the power to KIM, enter the PROM programming routine and the appropriate data. Make sure that the programming routine and the data have been entered correctly.

3. Enter the low address of the data into 00EA, 00EB, and the high address into 00EC, 00ED. The program will enter the cycle count into 00EE and use 00E8, 00E9 as the data-byte pointer. (00E7 is used as a false address during the program, so the contents will be lost.)

4. Turn on power to the programmer and examine the contents of peripheral port A (1700). With a blank PROM in the programming socket the data byte should be FF. Go to 0000 and hit GO. The programming will take just under two minutes. When the programming finishes the display will be 0000-A0.

If there was an error detected during the verification routine, the display will light up with the address of the error. To verify

A kit for building Jim's EPROM Programmer is available from:
O.C. Stafford Electronic Service & Development Co.
427 South Benbow Road
Greensboro NC 27401

	Drilled	*Undrilled*
PC Board (Part number KB/JHG 11478)	$ 7.50	$5.80
Parts kit (no board)	43.00	
Negative or positive	3.50	

Add $1 for shipping.

```
          Zero page registers:

00EE cycle count
00EC, 00ED high data address: HDAL, HDAH
00EA, 00EB low data address:  LDAL, LDAH
00E8, 00E9 data pointer: DPL, DPH
00E7  Dummy counting address

      Program:

0000  A0  00              LDY    #00                       Clear Y
0002  A9  64              LDA    #64                       Load cycle count
0004  85  EE              STA    EE                        and store 00EE.
0006  A9  FF              LDA    #FF                       Set I/O Port B reg. to
0008  8D  02  17          STA    DRB                       high state and I/O
000B  8D  01  17          STA    DDRA                      port A to output.
000E  A9  EF              LDA    #EF                       Set I/O Port B to output
0010  8D  03  17          STA    DDRB                      except pin PB-4.
0013  A9  F5        A1    LDA    #F5                       Clear programming
0015  8D  02  17          STA    DRB                       counters.
0018  20  6E  00          JSR    Mov                       Go Sub MOV.
001B  B1  E8        A2    LDA(Y) Data                      load data using data pointer.
001D  20  77  00          JSR    PRGM                      Go Sub PRGM.
0020  20  90  00          JSR    INCA                      Go Sub INCA.
0023  90  F6              Bcc    A2                        If less than or equal
0025  F0  F4              BEQ    A2                        to high Data Pointer
                                                           continue entering data.
0027  A9  10        A3    LDA    #10                       Bit test for end of
0029  2C  02  17          BIT    DRA                       cycle.
002C  F0  07              BEQ    A4                        if done go to A4.
002E  C6  EE              DEC    EE                        decrement cycle count
0030  D0  E1              BNE    A1                        and if not zero go to A1.
0032  4C  3D  00          JMP    VRFY                      Go to VRFY.
0035  A9  FF        A4    LDA    #FF                       Program FF into
0037  20  77  00          JSR    PRGM                      remainder of PROM.
003A  4C  27  00          JMP    A3                        Go to end of cycle test.
003D  A9  00        VRFY  LDA    #00                       Set I/O Port A as
003F  8D  01  17          STA    DDRA                      input.
0042  A9  FD              LDA    #FD                       Clear programming
0044  8D  02  17          STA    DRA                       counters with CS low :
0047  20  6E  00          JSR    MOV                       Go Sub MOV.
004A  AD  00  17          LDA    DRA                       Load data word from PROM.
004D  D1  E8        B1    CMP(Y) data                      compare to data word
                                                           in RAM.
004F  D0  12              BNE    B2                        On error go to B2.
0051  A9  FF              LDA    #FF                       Toggle the
0053  8D  02  17          STA    DRA                       programming
0056  CE  02  17          DEC    DRA                       counter.
005E  F0  EA              BEG    B1                        go to B1.
0060  4C  4F  1C          JMP    KIM                       Return to KIM program
0063  A5  E8        B2    LDA    DPL                       on error set
0065  85  FA              STA    KIM     pointer low       KIM pointer to
0067  A5  E9              LDA    DPH                       error address
0069  85  FB              STA    KIM     pointer high      and return to
006B  4C  4F  1C          JMP    KIM                       KIM program

      Subroutines:

006E  A5  EA        MOV   LDA    LDAL                      Move low data
0070  85  E8              STA    DPL                       address into
0072  A5  EB              LDA    LDAH                      the data
0074  85  E9              STA    DPH                       pointer and
0076  60                  RTS                              return.

0077  8D  00  17    PRGM  STA    DRA                       Output data to PROM.
007A  E6  E7              INC    Dummy Add.                Delay for data setup.
007C  A9  03              LDA    #03                       Raise PGM pin to
007E  8D  02  17          STA    DRB                       26 volts.
0081  20  8A  00          JSR    DELAY                     Go Sub DELAY.
0084  A9  06              LDA    #06                       PGM pin returned to GND
0086  8D  02  17          STA    DRB                       and counter incremented.
0089  60                  RTS                              Return.
008A  A2  F8        DELAY LDX    F8                        Delay for .99 msec.
008C  CA            C1    DEX                              (992 clock cycles)
008D  D0  FD              BNE    C1                        and
008F  60                  RTS                              return
```

```
0090  E6  E8         INCA   INC   DPL                      Increment
0092  D0  02                BNE   D1                       data pointer
0094  E6  E9                INC   DPH                      and
0096  A5  E9         D1     LDA   DPH                      compare to
0098  C5  ED                CMP   HDAH                     High Data
009A  90  04                BCC   D2                       address
009C  A5  E8                LDA   DPL                      and
009E  C5  EC                CMP   HDAL                     return
00A0  60             D2     RTS

        PEEK routine:

00A1  A9  00         PEEK   LDA   #00                      Set I/O Port A
00A3  8D  01  17            STA   DDRA                     as an input.
00A6  A9  FF                LDA   #FF                      then toggle
00A8  8D  02  17            STA   DRB                      the programming
00AB  CE  02  17            DEC   DRB                      counters.
00AE  A9  00                LDA   #00                      Set KIM pointer
00B1  85  FA                STA   KIM    pointer low       to I/O Port A
00B2  A9  17                LDA   #17                      and jump
00B4  85  FB                STA   KIM    pointer high      to KIM program
00B6  4C  4F  1C            JMP   KIM                      to display PROM data.
```

Program A. The programming program. The PEEK routine is used by entering A1 into 17FA and 00 into 17FB. Go to address 1703, hit the data key and enter C,F,F. This will clear the programming counters. Then go to address 1700. The data word displayed should be the first data byte in the PROM. By hitting the stop key, the PROM address can be incremented, and the data displayed will be the contents of the PROM.

the error load A1 into 17FA and 00 into 17FB. Then go to 1700.

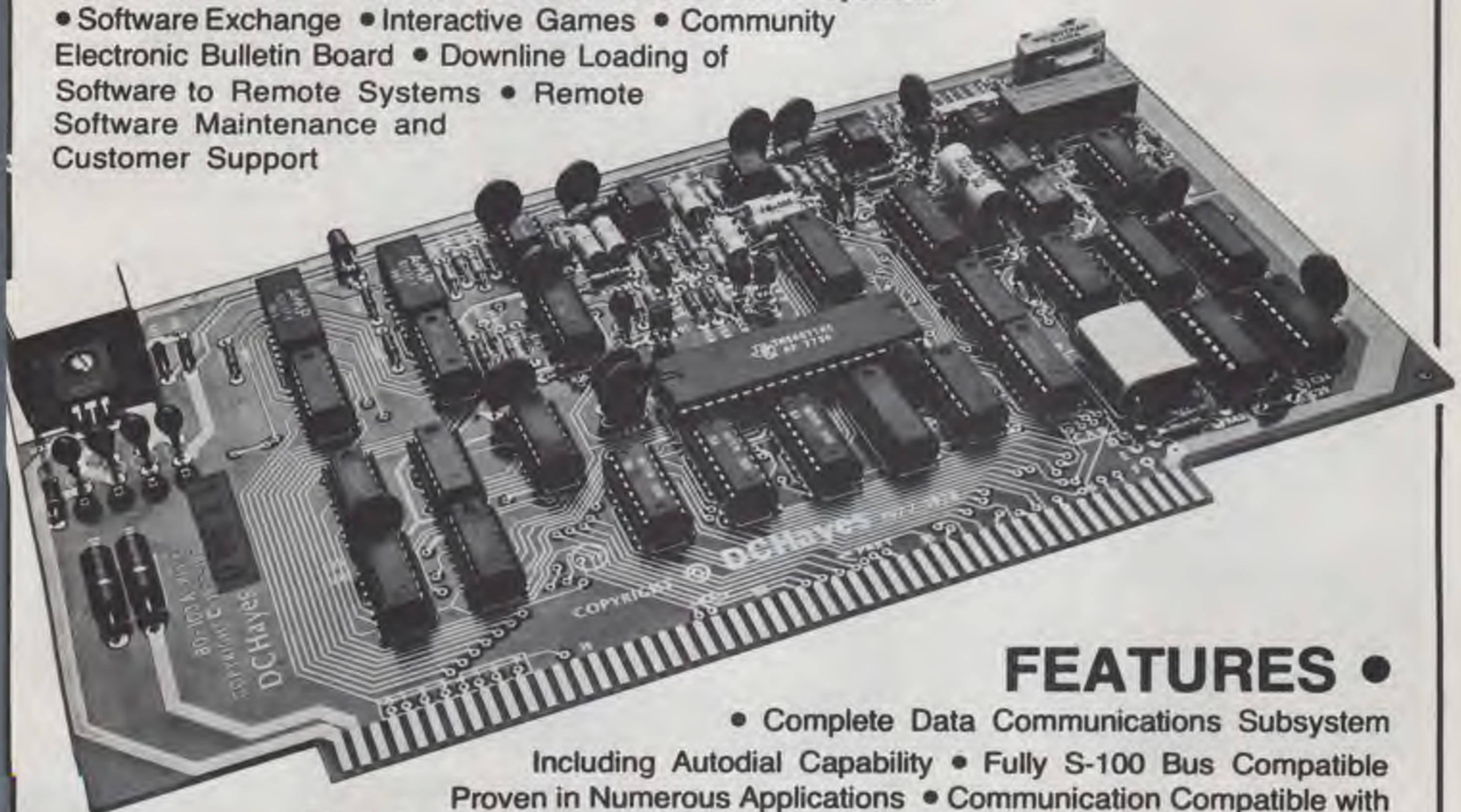
The data should be the first byte in the PROM. To increment the PROM address, hit STOP. The data should now be the second byte in the PROM. Step the PROM to the location of the error.

If the error is a high bit in the PROM (e.g., a 1 in place of a 0), the programming can be attempted again. If it is a 0 in place of a 1, turn off the power and erase the PROM. Then verify that the erasure has returned the problem bit to a 1 and repeat the programming. Always be sure to turn off the power before inserting or removing the PROM.■

modem / 'mo • dəm / [**mod**ulator + **dem**odulator] *n* - *s* : a device for transmission of digital information via an analog channel such as a telephone circuit.

USES •

• Intelligent Terminal • Timesharing • Distributed Processing • Automatic Data Collection • Access to Remote Data Base • Line Concentrator • Telecommuting • Electronic Mail • Remote Access to Process Control Systems • Software Exchange • Interactive Games • Community Electronic Bulletin Board • Downline Loading of Software to Remote Systems • Remote Software Maintenance and Customer Support



FEATURES •

• Complete Data Communications Subsystem Including Autodial Capability • Fully S-100 Bus Compatible Proven in Numerous Applications • Communication Compatible with North American Standard Bell System 103 Modems • Software Control of Originate and Answer Format, and Parity • Extensive Self-Test Capability • All Digital Modulation and Demodulation Means No User Adjustments Required • On-Board Crystal Oscillator Insures Proper Timing in all S-100 Systems • Break Generation • Meets or Exceeds all Bell System and FCC Specifications for use with a CBT Coupler (Data Access Arrangement -DAA) • 90 Day Limited Warranty

D.C. Hayes Associates, Inc. H20
16 PERIMETER PARK DR. SUITE 101
P.O. BOX 9884 ATLANTA, GEORGIA, 30319 (404) 455-7663
DISTRIBUTED IN CANADA BY TRINTRONICS, LTD., TORONTO

Heinz Gf. Matuschka
583 Tarragon Dr.
San Rafael CA 94903

Something Extra in Mass Storage

The speed and capacity of a dual phi-deck system should be considered when you decide on the mass storage for many serious applications. Meca's operating system and voice capability could constitute a deciding factor.

The author's system consisting of a SOL-20 with 32K of memory, the Alpha-1/2 with dual tape drives and a 19-inch Zenith TV, modified for direct video input.

Finding a highly reliable, efficient and low-cost mass-storage system has been a high priority with me since I got my SOL-20 up and running in March 1977. Let me define that last statement. By highly reliable I mean the ability to save a program with the assurance that you will be able to recover it. Even double saves on audio tape with the CUTS tape interface is not always a guarantee of program recovery. Speaking with owners of other similar systems, like the Tarbell tape interface, corroborates my observation concerning these devices.

Even though 1.2 kilobaud (or 150 characters per second) seems fast compared to typing or to some of the audio tape interfaces used by many microcomputer users, it is definitely too slow for anything but pure, and limited, hobby applications. A single-spaced typewritten page (8.5 × 11 inch format) contains 2700 to 3000 characters. This means that the simple transfer of one typewritten page will take between 18 and 20 seconds, which I consider unacceptably long.

Finally, from a cost point of view, the audio recorder storage is hard to beat, particularly if you already have the tape recorder. However, as stated above, for small business application an audio recorder is inadequate. Floppy disks appear to be the obvious answer, but that solution has drawbacks—the biggest of which (to most of us, anyway) is cost.

At present a dual floppy costs $2000 to $2500; a little more than half that much for a single. The micro or minifloppies have an almost 2 to 1 cost advantage over that, but only ⅓ the capacity. I'll get back to some more comparisons later.

Given the above, the ideal solution would appear to be some type of tape system that would allow the use of easily and inexpensively obtained audio tapes, but with data-storage capability and reliability now most commonly associated with floppy-disk storage.

The Rainbow's End

I checked with manufacturers of digital tape-recording units and systems. Some quoted delivery dates four months into the future; others told me how some of their *customers* were developing software for their systems, which they (the manufacturers) hoped to be able to offer soon; others not only let you worry about your own software, but even expected you to do some hardware modifications.

Frankly, I was seriously considering home brewing something, even though I had seen some efforts along that direc-

tion by other micro owners that had left me less than enthusiastic, considering the amount of effort it had involved for the individuals and the total result achieved.

I had already made quite a few phone calls in collecting the information I considered necessary to make an intelligent decision when I finally contacted Meca, the manufacturers of the Alpha-1. Now, thinking back about why I waited to check with Meca, I have to state that the picture Meca used for their advertisements for the Alpha-1 does not do justice to the unit, not to mention its capabilities. I placed the order for the Meca Alpha-1/2 (the second number stands for the number of drives) on Thursday, and received the kit by UPS the following Monday.

The assembly of the unit was straightforward; the assembly of the drive cards, one of which works with and is mounted on each drive, gave me the most problem. Not only are the drive cards not soldermasked, but also one of the drive cards had not been etched to Meca's specifications by the PC board manufacturer. Since I assembled my unit, Meca has started to offer their kits with the drive cards assembled and tested for an extra $20 per card. For my money, particularly knowing how much work went into my drive cards, I would gladly pay that amount if I had to do it all over again.

Assembling the Meca Alpha-1

The total assembly time came to just over 30 hours (including the Meca drive cards, which represented 8 to 10 hours). However, there were some minor difficulties I would have run into even with the assembled cards. Most of my problems occurred with the drive card installed on drive 1, which has one mounting screw that is most inconveniently blocked by the power transformer. After most of the problems had been fixed, I finally switched it to the 0 drive.

The S-100 bus interface card, fully soldermasked, could only have been improved by silkscreening the information in white, rather than black, for better legibility. One of the 7805 voltage regulators, lacking any printed information on the body, turned out to be a 2.5 V regulator instead, which caused some rather interesting voltage readings until the problem was corrected. In getting my system ready to run, I called Meca and found them both eager to help and friendly at all times. They urged me several times to return the poorly etched drive card to them for repair or replacement.

Meca, who advertise that their Alpha-1 mass-storage system is ready to use with Mits, Imsai, Poly 88, Z-2 and ZPU, finished the software for the SOL-20 with SOLOS just as I was completing my assembly, so there was no delay in my being able to bring the system up (and which, in all probability, made me the first SOL owner to operate with the Meca).

Included in the price of the unit is the relocatable Meca cassette operating system (MCOS), complete with an assembler, an editor, debugging program, a directory reconstruction program and a patch to allow the MCOS to operate and cooperate with the 4.0 Altair extended as well as with one of their 8K versions of BASIC. (Meca has since become distributor for the Microsoft 4.4 extended and 8K versions of BASIC, which are the same as Altair BASIC, and will sell them to their customers for $150 and $75, respectively.)

The Software

The following two weeks comprised a fascinating voyage of discovery. Never having had an assembler to play . . . excuse me, to work with before, I loaded some of the source codes Meca supplies with their software and let the assembler assemble it on screen. The assembler is a modified version of the one supplied free of charge by Processor Technology. It is patched to display 15 lines on the screen at a time, with 'Return' bringing successive 15-line pages on screen.

When using the Meca Editor, first load it into memory. When you then load in a program to be edited, the MCOS keeps track of where that program was loaded, and upon giving the "edit" command, the editor will automatically go to the start of the program to be edited.

The Alpha-1 shown after assembly of the drive cards, the interface card and the LED and toggle-switch harness installed in the cabinet top. *(Photo by Milt Hyman)*

The editor allows you to delete one or any number of lines simply by quoting the beginning and ending line numbers of the block to be deleted. There is an insert mode, a replace mode and an add mode. When adding lines, the editor will number consecutive lines automatically.

One problem I ran into when experimenting with the editor and assembler was (and I am certain that this is applicable to any other editor-assembler combination) that when I put my own test programs together and attempted to assemble them they crashed every time. It took me a day to find out that the assembler, which requires an end statement to know when it has reached the end of the assembly-language source listing, wanted that end statement, not in the label field, where I had innocently placed it, but in the op-code field.

I make specific mention of this here because I just got a call from another computer owner who bought an Alpha-1 on my recommendation and was stumped by the identical problem. I am sure that most computer professionals would get a chuckle out of such an episode; however, to me, it typifies the kind of frustration the novice computer owner/operator can fall victim to. Simultaneously, it is the type of problem a professional couldn't solve because more often than not he takes for granted that anyone involved with computers possesses such "elementary" information.

The debugging program will allow you to initiate a hex dump from any place in memory. It also allows you to enter new data directly in hex, and, in one of its phases, will automatically display the contents of the CPU's registers.

As you lose interest in certain programs within a tape, you may delete them from the directory. On its way to the beginning of programs requested from the drive, the unit will skip over these vacant spots at up to 120 inches per second (ips). It is possible, however, to recover any deleted files via the Directory Reconstruction program. In that mode the unit will go through the tape at read speed (5 ips), displaying the name of the program, the number of bytes in it, the starting address on the tape, the starting address of the program as recorded when the program was first written (sometimes these two addresses may be different by

several counts), and announce if it happens to be a duplicate file name.

Still More to Come

As I was still attempting to take in all the goodies that had been supplied with the Alpha-1, I was informed by Meca that they had just completed an updated and improved MCOS, designated version 3.0, which they would initially offer to owners of Alpha-1 and individuals buying units for $10 ($15, if acquired at a later date). I couldn't imagine how, or what, they could have improved on a system which at this point had already by far surpassed what I had expected. But they did it!

Previously, after loading MCOS you had to *rewind* the tape, *read* the directory and *list* the directory to determine the contents of a tape. Version 3 automatically mounts the home drive, which simplifies start-up. Version 3 defaults to the last drive used when not specified. Everything has been made even more forgiving than it was; the equipment virtually does all the housekeeping for you. When you really think about it, equipment like this really should. After all, why do you have a computer if you have to do all its thinking?

The Alpha-1 with Altair BASIC

If your present BASIC does not let you do *whatever* you can think of (with a computer!), and make it easy to do, you owe it to yourself to look at the Altair BASIC; and I mean before you spend a lot of time writing potentially valuable programs in something less versatile.

What has made the Alpha-1 valuable to me over and beyond all I have mentioned before is that the new version 3.0 MCOS can be grafted on top of BASIC (Altair) in a fashion that lets BASIC look through it at all the free available memory. This means not only that you have all of the BASIC functions at your beck and call, with program CSAVE and CLOAD working directly with the Alpha-1, as do the *array* CSAVE and CLOAD features; you can also use the five-character name acceptable to the Alpha-1 as a reference and file name (as opposed to the single character the BASIC requires). You are further able, either by direct command *or* in a BASIC program, to use all of the pure Meca Alpha-1 commands by placing them after a REM*.

Furthermore, the new MCOS allows overlays of existing programs so that you can make desired changes to a program and then save the altered program into the same slot that was previously occupied by the original version. If the altered program is less than one kilobyte larger, it will be stored in the same slot; otherwise, the Alpha-1 will automatically add the altered file at the end of all the existing files and delete the old file. Thus, you can call in data, update it, change it, then write it back in place . . . similar to a disk operation.

Some Alpha-1-to-Disk Comparisons

Just one week ago I was informed by the local Byte Shop in San Rafael CA that a 300K byte disk (a Processor Tech Helios II) can be copied in about six minutes. Frankly, I was amazed. I had always been under the impression that it would take a half to one minute, at most!

Now, comparing that to the Alpha-1, I calculated the following: The Alpha-1 has a data transfer rate (read or write) of 781 plus a fraction bytes per second (6250 baud). In the copying mode, it reads a file into memory, writes the file, rewinds and reads the file back, checking for errors.

When the read-back check is completed, the unit is ready to load the next file to be copied. In effect, for each file copied, the Alpha-1 makes three passes—two read and one write. Since this reduces its effective copying speed to a "mere" 260+ bytes per second (still think your audio cassette is fast?), it would take 1000 seconds to copy 0.26 megabytes. Allowing for some extra time for the search runs, you can copy the same amount of information it takes the disk six minutes to complete, in 18 to 20 minutes with the Alpha-1. Therefore, the disk is three times as fast as the Alpha-1! (Just remember, it also costs three times as much.)

Now, I don't want to convey a sour-grapes attitude . . . if you *need* the speed, I am quite willing to concede that a disk is, presently, a viable, if expensive, solution. But let's get back to the Alpha-1. Here, with a dual drive unit you get over 50 percent more data on line than with the full-size dual floppy. And if data on line becomes critical, as when you want to run a complex sorting system (possibly with the computer put to the task overnight), you might even prefer 2 megabytes on line. You could manage this with a four-drive Alpha-1/4 for about two thirds the money you would otherwise pay for a dual-disk drive, which only offers one third of the Alpha-1/4 capacity (on line).

Another way to increase on-line storage without buying a second Alpha-1/2 (dual drive) is to use 90-minute cassettes, although this would increase worst-case access time by 50 percent. With greater difficulties (i.e., with some hardware changes), you could conceivably use the second track, which is reserved for an analog channel (sound), as an additional digital information medium.

Since you operate on larger blocks of data with an Alpha-1 than with a disk system, your requirements for RAM memory will be higher. Of course, with

The fully-assembled Alpha-1/2 (see text) with optional wood side panels. *(Photo by Milt Hyman)*

memory prices dropping so rapidly this becomes desirable since RAM is the heart of your system.

Now let's look at some other important differences. The very speed of the disk can bring about utter disaster in the event of a head crash. I understand that if that situation occurs, you can lose all the information stored on your disk in the blink of an eye. With the wrong type of glitch fouling up your tape-system equipment, you might lose some data *if* the glitch turns both the tape transport *and* the write mode on simultaneously and puts the transport into play mode.

The mini or microfloppies have been hailed as a great boon to the microcomputer owner because they go for about *half the price* of a full-size floppy. Of course, on-line storage is cut to one-third of a full-size floppy. If the program you want to run has less than 90K bytes data requirements, you better your access speed (for 90K only) from about six seconds for the tape drive to half a second on the microfloppy. But if you need as much as 10K bytes more than you can put on the disk, you're cooked.

Some Potential Major Side Benefits

The Meca Alpha-1 was designed with some important performance goals. For reliability the Alpha-1 uses the industry standard phase encoding method and uses a professional type phase-locked loop to smoothly track data over a 2:1 speed range. This allows tapes made on one Alpha-1 to be totally compatible for use on any other Alpha-1, without adjustments of any type. It further means that although *you* may have an Imsai or Mits and *I* have a SOL, the programs we write or develop, using Altair BASIC, can be exchanged between us directly and will run without changes on each other's machine, since the different I/O routines of our respective equipment are effectively taken care of by our own versions of BASIC.

It also makes an *intercomputer* Alpha-1 users group feasible. As a member, you could benefit from efforts expended by individuals in writing programs on computers other than the type you own, yet with whom you'll be able to exchange software as simply as exchanging tapes. Call it another breach in the software bottleneck.

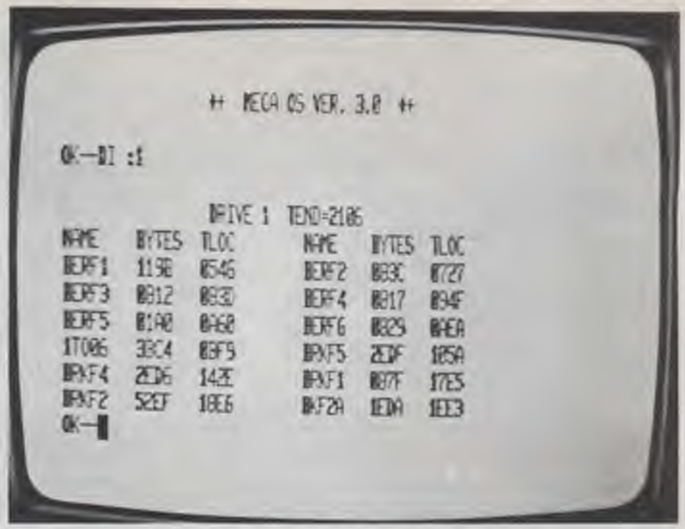
In Conclusion

It is my understanding that Meca will have an excellent text editor working with the Alpha-1 soon, even though, since this is a royalty arrangement with the author of the text editor, you will have buy this piece of software. Other than that, Meca appears to be working on a CCD memory buffer between the Alpha-1 and the computer to narrow the gap to the disks in those areas where the disk does have an edge.

At this point, you probably think that I am sold on the Alpha-1. Well, you are right—I am. And so, incidentally, are a number of individuals who purchased their Alpha-1s based on my recommendation; the software exchange has already started.

The thought that I may soon be able to exchange software with you, who are reading this, adds to my feeling of satisfaction at having acquired this versatile unit.

Finally, if you think that this might be for you, but even $695 (plus tax and shipping) is a little steep for you, Meca also sells the Alpha-1 as a single-drive unit (under $500). This robs you of the advantages of the double drive, such as automatic drive-to-drive copying, but it will still be at least a couple of orders of magnitude better than the ol' audio cassette.■



Screen display of a 12-file directory. If the directory is bigger than 28 files (50 or more are possible), the display stops, waits for a C/R to display the next page.

William L. Colsher
4328 Nutmeg Ln., Apt. 111
Lisle IL 60532

From Big BASIC to Tiny BASIC

Don't tear out your hair if your BASIC seems lacking; tear into this article.

If you've been operating your system for any length of time, you've probably run up against the following situation. While perusing game programs in a book or magazine, you see one that really blows you away; you just have to have it on your system. A closer look results in disappointment: Your version of BASIC lacks a feature essential for the program. You sigh and wonder if you'll ever get a better version of BASIC and curse the authors of your present BASIC.

It's not nice to swear; if your mother heard you, she'd probably faint. So take heart, there are ways around some of those problems; this article takes a look at a few of these solutions.

The Array Problem

There are quite a few programs that make use of two-dimensional arrays; Klingon Capture, Star Trek, Go Maku, Ticktacktoe and Wumpus are just a few. Some programs, like Depth Charge and Qubic, use *three*-dimensional arrays. These are some of the neatest games around. How will you tell those neighborhood kids who have been begging to see your system that they can't play Wumpus because you can't program it? What to do? The following discussion is guaranteed to keep you, and the kids, happy.

Let's assume you wish to use a program that makes use of a two-dimensional array. Somehow, you must turn it into a one-dimensional array that your version of BASIC can handle (this also applies to changing three-dimensional to two-dimensional arrays). First, let's see how an array appears to BASIC. When the BASIC interpreter receives a DIM statement, it looks at the expressions in parentheses that accompany the variable names. Based on those expressions, it then reserves storage for the array elements.

Fig. 1 shows a DIM statement for a ten-by-ten array (called A). Clearly, this array has 100 (10 * 10) elements, so a one-dimensional array of 100 elements should be its exact equivalent. The only problem is how to index into it. We can't just multiply the two subscripts (M and N) since the product wouldn't be unique. For example, element A(1,2) and element A(2,1) in the old array would both occupy the same location in the new, one-dimensional array.

If we think of an array DIMensioned M by N, we can imagine it as being M groups of N locations. When we use a pair of subscripts, we are actually specifying which group and which member of that group we want. In order to save space (you'll see why we do this shortly), let's start numbering the *groups* with zero. Thus, we'll

```
10 DIM A(10,10)          M = 10, N = 10
   .
   .
   .
90 LET A(X,Y) = A(X,Y) + 176
   .
   .
   .
To convert two-dimensional subscripts to one-dimensional,
use this formula -((X-1)*N + Y),
where -X and Y are the subscripts and N is the second number
  in the DIM statement.
The program segment becomes:
10 DIM A(100)
   .
   .
   .
90 LET A((X-1)*10 + Y) = A((X-1)*10 + Y) + 176
If, in the above example, X = 7, and Y = 3, we would have:
  A(7,3) equivalent to A((7-1)*10 + 3)
                       A(63)
```

Fig. 1.

```
100 LET R=RN          RN is a BASIC random-number function
110 LET T=R
120 LET R=R/N         N is the modulus of the random # we want
130 LET R=R*N
140 LET R=T-R+1       Gives random # R where 1 ≤ R ≤ N
```

Fig. 2a.

```
ORIGINAL CODE:   150 LET R=RND(0)
                 160 IF R<.1 THEN GOTO 500
TINY BASIC EQUIVALENT:  150 GOSUB 1000*   Get a mod 100 random # R
                        160 IF R< 10 GOTO 500
     *Assume that line 1000 starts a subroutine like 2a with N=100.
```

Fig. 2b.

have group 0, members 1 through N, and so on, all the way up to group M-1, members 1 through N.

Suppose that the subscripts for array A in our original program were X and Y and that the array has been DIMensioned M by N. Now, since we want to start with group 0, we'll need to subtract 1 from the X subscript wherever it appears to specify the group. Since the groups have N members each, we'll also need to include that factor (the N) as well as the Y subscript to tell with which member of the group we're dealing.

Fig. 1 lists, and provides an example of, the formula for converting two-dimensional array subscripts to one-dimensional if you take all those factors discussed above into account.

Modulus Number

Random numbers, a big part of computer games, generally have to be in a particular range. If you have an eight-by-eight grid, you don't want to come up with numbers like 163 and 59; you want numbers between 1 and 8. The range of a random number is called its *modulus*. So, for an eight-by-eight grid, we need a modulo-8 random number.

Many BASICs generate random numbers in the range of zero to one ($0 \leq R \leq 1$). Getting that into a range you can use in a game is quite simple . . . just multiply by the modulus number (8 in the example we've been using). Some BASICs, though, generate random numbers in a different range, 0 to 32767, for example. Obviously, that doesn't do us much good if we want a number between 1 and 8.

In division the remainder is always in the range of 0 to D-1 (where D is the divisor). This should give you a clue as to how we're going to get our random number. Fig. 2a shows how to get to that remainder in BASIC. This particular example is with a version of BASIC that only has integer arithmetic; so if yours uses floating point, you'll have to make use of the INT function in all the calculations.

Many times programs use random numbers to determine if something good or (more likely) bad will happen to a player. Since they are using numbers between zero and one, and want a ten percent chance that a given thing will happen, some code such as that in Fig. 2b should be provided.

Replacing ON-GOTO

Programs, particularly games (of course), often use the ON-GOTO feature of BASIC. However, this extension is not available with many versions. The obvious way to get around this problem is to replace the ON-GOTO with several IF statements. This lacks elegance and also uses an incredible amount of storage; but with a little judicious line numbering, it is possible to form a very short replacement for the ON-GOTO.

This technique requires that the various segments of code must all be numbered in some way to be calculated more easily and uniformly. For example, they might start on hundreds (see Fig. 3). If you can't renumber the program to match a pattern, you can place several GOTOs in the program to get to the required location. This is not as elegant, but it is better than several IF statements, and it makes it easy to add new commands.

```
ORIGINAL CODE:      100 ON X GOTO 1000,1100,1200,1300,1400
TINY BASIC CODE:    100 GOTO 900+(X*100)

     The GOTOs Technique

100 GOTO 5000+X-1
 .
 .
 .
5000 GOTO 126
5001 GOTO 713
5002 GOTO 2010
5003 GOTO 1075
 .
 .
 .
This is the equivalent of:
     100 ON X GOTO 126,713,2010,1075
```

Fig. 3.

There is one major problem with using an expression in a GOTO or GOSUB statement: The line renumbering function of most versions of BASIC will choke on it and foul everything up. Here's what generally will happen. A number starting the expression will be changed to correspond to whatever it should be, and the rest of the expression will be left alone. The relationship between the sections of code that this technique depends on will be destroyed in any case. If you enclose the entire expression in parentheses, it will be unchanged, but once again, the relationships down in the program will be fouled up. This will also happen if a variable, instead of a number, begins the expression. The answer to this problem is not to use the renumbering function.

One of the most aggravating occurrences in a program is to come across two-character variables (i.e., alpha characters followed by numbers) if your version of BASIC doesn't allow them. The answer to this problem is incredibly simple: DIMension an array with the alpha character as its name and use the numeric portion of the original name as a subscript. If your BASIC doesn't allow zero as a subscript and it is used in the original program, just add one to all the subscripts.

```
ORIGINAL CODE:   100 LET B1=((B2*C3+B3)/C5)+B4
NEW CODE:          1 DIM B(4),C(5)
                   .
                   .
                   .
                 100 LET B(1)=((B(2)*C(3)+B(3))/C(5))+B(4)
```

Fig. 4.

Fig. 4 illustrates this technique. You may waste some space in unused array locations, but if you want to run the program . . .

You'll also find that building an array will be useful in programs that use a lot of variables. When it appears you'll run out of letters and you can't use those two-character names, just DIMension an array to whatever size you need and you have that many more variables to use (T for Temporary is a good choice for the name).■

from kilobaud:

Instant Software

only $7.95 each

THE FIRST SIX PROGRAMS:

•PERSONAL WEIGHT CONTROL/ BIORHYTHMS—SP0105—by Microcosm (for the PET™) Here's a program that will let you chart any individual's biorhythm cycle, showing physical, emotional, and intellectual highs and lows. If your neighbors find out you've got it, they will never leave you in peace. **PERSONAL WEIGHT CONTROL** is a must for the diet-conscious. Once your ideal weight has been determined, the program will calculate the correct daily caloric intake necessary to reach your goal over a certain time period. $7.95.

•SPACE TREK II —SP0102—by Ed Juge (for the TRS-80™) This Space Trek game will challenge the most skillful and careful player. As commander of the Enterprise, you must defend the galaxy against enemy Klingon warships using your arsenal of phasers and photon torpedoes to best advantage. The Enterprise is also equipped with impulse power for moving within the quadrant, and warp power for jumping from quadrant to quadrant. $7.95.

•BEGINNER BACKGAMMON/KENO—SP0104—by Fred Schild and Ed Juge (for the TRS-80™) You wll never be in need of a backgammon companion again! Schild's program is designed for the newcomer to the game, but the computer's steady play will challenge the careless expert. **KENO** is a popular Las Vegas gambling game. As in many games of chance, the odds are against you, but when you win you can win big. $7.95.

•BASIC LUNAR LANDER/INTERMEDIATE LUNAR LANDER—SP0101—by Ed Juge (for TRS-80™) Your automatic landing system has malfunctioned and you are forced to make a manual landing. Taking control at 15,000 feet, the pilot's success requires precise manipulation of the lunar lander's retro-rockets. After mastering the basic game, the able commander will be taxed by the difficult intermediate program. $7.95.

•MORTGAGE WITH PREPAYMENT OPTION/FINANCIER—SP0106—by Microcosm (for the PET™) Before you go to the bank, calculate your mortgage payment schedule and find out what you can afford. You will learn how much prepayments save, and can easily compare the true costs of variations in interest. **FINANCIER** is actually three programs in one and is guaranteed to make you a financial wizard. In part I, you'll learn to tell which investments will pay off. Part II calculates annual depreciation schedules, providing rates, amounts, and salvage values, which is great when tax time rolls around. The final section figures costs of borrowing, payment terms, and remaining balances on loans of all sizes. $7.95.

THE KILOBAUD PACKAGE—CASSETTE PLUS BOOK Each Kilobaud software package includes an informational booklet as well as a cassette. The book provides a hard-copy version of the program as well as information about the program's operation, design, and intended use.

KILOBAUD QUALITY CONTROL Kilobaud cassettes are produced in the same audio lab that has been turning out thousands of ham radio code tapes each month, with an enviable record of quality control. To make sure that Kilobaud's cassettes establish a high level of quality in the field, each tape is individually tested, and the best tapes available are used .

MORE TO COME . . . As of this printing, over 100 different programs have been tested in the Kilobaud lab, and are being readied for publication. Use the coupon below to order our FREE software catalog. You'll be able to get prerecorded programs in business, electronics, hobby games, self-study including: small business, accounting, checking account/general ledger, the slot machine game, home checkbook handler, a simple mailing system, home weatherman, personal stockbroker, the missle game, and many others.

☐**Please rush me the following cassettes:**

☐**Send me your FREE software catalog**

For Toll Free Ordering Use Your Credit Card And Call 1(800) 258-5473

kilobaud

peterborough nh 03458

Qty.	Catalog #	Description	Unit Price	Total
			Add $1.00 shipping & handling for orders under $10.00	
			Total:	

Coupon expires in 60 days.

Enclosed $______ ☐Check ☐Money order ☐Bill me (subs only)

Bill my: ☐American Express ☐BankAmericard/Visa ☐Master Charge

Credit card #__________

Expiration date ______ Signature __________

Name __________

Address __________

City __________ State ____ Zip ______

KB 9/78

TROUBLE-SHOOTERS' CORNER

(from page 11)

practically identical. The VIM drives (successfully) the same VIA we've been discussing.

Now let's take a closer look at Steve Wozniak's use of Ø1. There are *two* Ø1s, and they're *not* the same. The real Ø1 comes out of pin 3 on the 6502, goes through gate C14, and gates the buffer amplifiers for the data bus—a critical timing function. This is a good straightforward design.

The other (bogus) line labeled Ø1 comes out of the $\overline{Q}$ (B1, pin 6) of the same flip-flop that drives the bogus Ø2 (really Ø0 and the gate delay of B11). This is the bogus Ø1 which appears on the interface bus at pin 38. It is really the inverse of the bogus Ø2 (pin 40) and is closer to being $\overline{Ø0}$ than the real Ø1.

I haven't checked all the places where this bogus Ø1 is used, but it *is* operating successfully on the main board and video, so it's probably OK. Since we have no intention of using it with the VIAs or PIAs, we'll let it off with a warning: *It has been determined that the use of Ø1 on the user bus may be dangerous to the health of any I/O design.* Since gate C14 is an LS device, the real Ø1 should be able to drive one or more additional buffers, if a new design should prove critical in this area. For now, we'll leave it—as is.

Some Troubleshooting Fundamentals

I said earlier that I had been looking for something like this with which to illustrate some of the theory of troubleshooting. Let's get into the fundamentals of "defining the problem." There are probably as many ways of defining a problem as there are *problems*—but the majority of techniques for troubleshooting hardware bugs fall into three methods and the combinations formed thereof.

The fastest and usually most practical is *substitution.* The next is *signal tracing.* These two are indigenous to nearly all electronic devices. A third (peculiar to computers) is hardware/software trade-off. Eventually, I expect to dedicate a whole column to each, but for now we'll take a little "preview of coming attractions."

Substitution. Substituting is even more effective in troubleshooting than on the playing field. It requires a minimum of knowledge and training, and can usually isolate a problem faster than any other method. It's my first choice whenever practical.

Substitution has two major drawbacks. The most common is that there is often no spare available to substitute. The second is best illustrated by the example of replacing a fuse. If the fault is still present, the "substitute" fuse is destroyed. This is a sobering thought when expensive or hard-to-get devices may become "fuses."

If I intend to *swap* chips or boards, I try always to check the supply voltages as close to the suspected fault as possible, *before* swapping. A solder bridge or loose bit of metal lodged between two device leads can zap a $50 IC in a split second. If the supply voltages are OK and the circuit was OK at some earlier time, then burnouts due to swapping on 5 V TTL/MOS systems are rare. On the other hand, if you've been fooling around with a soldering iron or reconnecting cables, then *beware!*

Signal Tracing. If substitution isn't practical, then your next best bet is signal tracing. If you've ever tried to follow a carnival con-man with his three-card monte or shell-game movements, then you have some idea of what it was like trying to trace out the results of substituting the real Ø2 for the Ø0 in the foregoing example.

Signal tracing usually falls into one of three general categories: signal detection (using scopes, pulse catchers, voltmeters, etc.), signal injection (using pulse injectors, oscillators, waveform generators, etc.), and a relatively new system of *signature analysis.* Signature analysis is too new (and expensive) for me to use, but as devices become more complex it may become the best practical way to accomplish in-field signal tracing. The other two techniques were used extensively in the example at hand.

The most powerful tool is the Hewlett-Packard #1615A logic analyzer used by *EDN.* I doubt whether many personal computing fans can afford one, or even have access to one (certainly I don't). On the other hand, a dual-trace oscilloscope is usually available in a metropolitan area if you have enough friends. A few years ago a voltmeter, VTVM (vacuum tube voltmeter) or DVM (digital voltmeter) was an absolute necessity for any kind of electronic troubleshooting, and when it comes to power supply and analog problems, they're still a *must.*

The advent of TTL and CMOS technology produced some new devices: the pulse-probes. They are rapidly becoming a must, and I hope we'll be doing a special article on them in the near future. If you have access to a good dc scope, it can usually duplicate the functions of a meter and/or a pulse probe.

The most common method of signal tracing is to put a normal signal into an input and follow (or trace) it from one device to the next using one of these tracers, until you find a place where it misbehaves. This can get rather complex, as the *EDN* article testifies. A not-so-often-used method is to inject a signal into the system and watch what comes out. We did this in testing the VIAs by building up a counter circuit that would provide a known pattern on the VIA input bus.

Hardware or Software?

Many troubleshooting aids can be implemented in either hardware *or* software. Most large corporate-development efforts (*EDN* included) are polarized into hardware- and software-type personnel.

Personal computing is breaking down this barrier, and one of the advantages of mixing these two disciplines is that the trade-offs between them can be used for troubleshooting. On the one hand, I mentioned the *hardware* counter we used to provide a known digital input. On the *software* side, we wrote loop programs that generated an alternate AA (1010 1010) and 55 (0101 0101) pattern so that we could trace through hardware with the scope.

The diagnostic system built into the PET is a giant step in the direction of using software to troubleshoot hardware. When people get tired of writing programs for games, maybe they'll put some effort into writing diagnostic programs directed towards isolating faults. Of course, the basic fallacy occurs when you try to use a faulty system to troubleshoot a faulty system.

Let me point out the spots where we used some of the specific, fundamental troubleshooting methods I've just described. By *substituting* Dave's PR-40 printer interface with its PIA, we were able to identify the problem of using POKEs from BASIC (more on that later). By generating test patterns with *software* loops, we were able to *trace* the bogus Ø2 waveforms and compare them using an oscilloscope. By *swapping* back and forth between the VIA and PIA and *signal tracing,* we established that, with a marginal

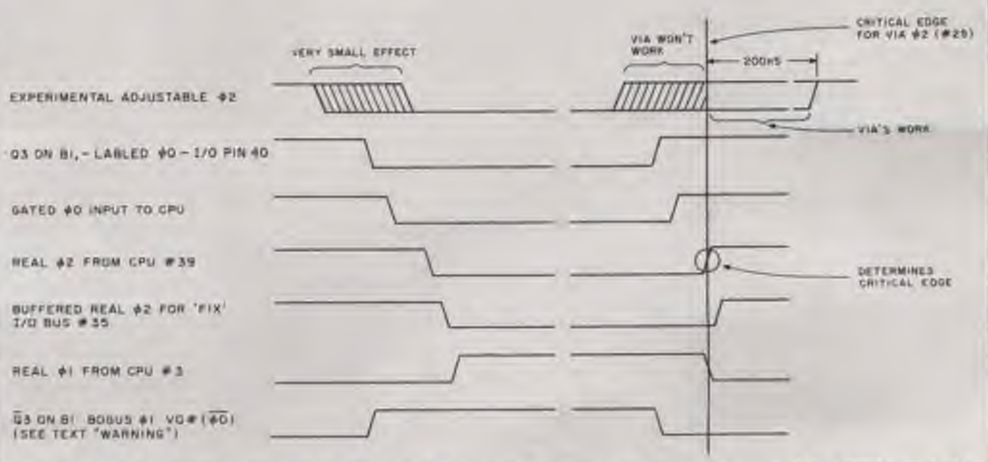


Fig. 3. One MHz clock timing diagram for Apple II illustrating errors uncovered by EDN and the Inmarco "fix."

system:

1. It's easier to write than read.

2. The PIA was less critical than the VIA.

3. Both worked if the rising edge of the driving pulse was delayed with respect to the real Ø2.

4. The 2708 memory would function on the same card as the PIA, even when the card was driven by different Ø2s.

5. The chances are very good that any other member of the 68XX or 65XX I/O devices can be expected to work on a modified Apple II, but we have no intention of *proving* it.

6. The "fix" for our Apple II consists of one TTL gate delay costing less than 25¢ (remember what I said about the *defining* being tough and the *fix* being easy?).

Things That Go Bump in the Night

Haunted houses, cemeteries and strange noises late at night are all part of a basic fear in mankind—the fear of the unknown—the *what-if* syndrome. Last weekend I was plagued with many fears. This weekend they're (nearly) all gone. There was still the problem of why we couldn't POKE into the PIA or VIA from BASIC.

After modifying the board for the real Ø2 buffer, we tried the BASIC program experiment again, with the same negative result. The VIA would work with machine code but not with BASIC, and the same double pulse showed up where there should only have been one. It really wasn't a problem for us because we expected to program in machine code, but it bothered me.

Synergistic Synectics

You may recall that I rated the synectic relationship of Dave Gordon as a top priority "asset." As it worked out, we didn't need this "tool" until Friday, but it was there and probably cut a full day off the debugging. As I said, Dave is neither an engineer nor a programmer, but he has assets that make these two skills redundant.

Dave is a gregarious, natural-born "horse trader" in the best American tradition. Personal computing is a fertile field for bartering, and Dave revels in it. His collection of programs and accessories for the Apple II is probably the largest in existence, but he "collects" something else with even more vigor: friends. He's up half the night with them, running new programs, expanding horizons, swapping "things" and information—all with an infectious, unbridled enthusiasm. If he doesn't meet enough people through the various computer clubs he belongs to, he'll have a barbecue at his home with the top brains in the Apple Corps.

This is what synergistic synectics is all about, and when an article like the *EDN* piece threatens their rock-solid belief that the Apple II is the best-thing-that-ever-happened-to-humanity, things happen.

The day after we started debugging, I found myself on the phone being introduced to Dave's friends, who were eager to help out. It turns out that Craig Vaughn and Sandy Tiedman had encountered the same problem and "fixed" it by using CMOS delays on the PIA boards just as we had done the first day—but they had *fixed the bug long before the* EDN *article was published.*

While discussing with Sandy what we'd found on Friday afternoon, I described the bug we'd observed in using the BASIC POKE command. He called back later to report that his Apple II worked fine with BASIC. On Saturday I had a long talk with Craig, and he agreed to try the same experiment. When we got back to software, we found our

own bug, and now our BASIC is home free. The point to be made is, how would you tackle a problem like this—without friends? How long would it take to solve it? How much manpower? How much equipment?

A Rebuttal

A "box" in the *EDN* article recounted all the shortcomings of the 6502 programming language. At the end, Mr. Jack Hemenway asked the rhetorical question, "Of course programmers can get around all of the above problems, but why should they have to?" Now let's be fair! He obviously preferred to program the 6800—so do I, but there *are* reasons why the 6502 is designed into so many computers.

Since I have eleven 6800 micros, an excellent 6800 program development system and a personal preference for 6800 language, why did I buy an Apple (6502) . . . even after reading the *EDN* article? Probably the most important reason is speed. A single byte register is faster than a double byte; it's a trade-off between hardware and software. If you want the speed, you write more code.

Hemenway made no mention of some of the 6502's pluses, such as the three-way branch capability after a bit test, the decimal-binary shift and the indirect address capabilities. No mention was made of the "sweet sixteen" ROM interpreter, which gives the Apple more double-byte power than any 6800 system I'm aware of. If you want to prove the speed difference to yourself, try benchmarking a program in BASIC (such as a bubble sort) on a PET (6502) and then on an SWTP, Altair 680 or Sphere (6800s).

I personally bought all of these machines. For my money, the 6502 can outperform the 6800 when it comes to speed, but it's not as easy to program. Do you design a microcomputer to please the end user or the programmer? I've been reading *EDN* ever since its inception, and I can't recall its ever being as editorially unfair as it was in this article.

Fan-atical Behaviors

If you've ever witnessed the spectacular behavior of a Dodger fan at World Series time, you've seen histrionics and heard exaggerations (from otherwise "normal" people) that make a used-car salesman look like a saint. In Europe, soccer matches can lead to bloodshed. If you've ever watched a PET lover battle it out toe-to-toe with a TRS-80 devotee, you've seen elements of the same phenomenon—but when the shouting's over, even the most dedicated *fan* knows that in Mudville even the mighty Casey *can* strike out.

I enjoy observing Dave and the members of the Apple Corps. I know that Dave realizes that the Apple has a design error in it. Even if it can be fixed for 25¢, any bug that requires the amount of troubleshooting effort that this one did is *serious*. He may verbally refer to it as a "little one," but he knows better, and (in private) admits it.

Rationality through Diversity

If *EDN* were to scrap the Apple-Indecomp project because of a "two-bit" defect, I, for one, would consider it grounds for serious criticism. *EDN* might well examine the factors that have made the Apple one of the largest selling micros in history! Can all those people be *that* wrong?

In delineating the shortcomings of the 6502, Jack Hemenway cries out to "move on" to a different microprocessor (probably 6800). I agree that *EDN* should move on, ***but not yet!*** The 6809 and 8086 *double*-byte CPUs are just around the corner, and I know of at least two houses that intend to put them into computers as soon as they're commercially possible. I heartily agree with *EDN*'s original choice of CPUs. In my opinion the Apple II was, and is, the best one for the Indecomp project (until the doubles appear).

Personal bias is one thing; responsible editorial bias in a major publication is something else. I repeat Bob Jones's statement: "If we all work together on this thing—we can all be successful, *together*." Synergism!

If you have interesting problems, solutions, comments, or indeed, anything you believe can help others, or that we can help you with in this area, drop a line to: Troubleshooters' Corner, *Kilobaud* Magazine, Peterborough NH 03458.

Publisher's Remarks

(from page 21)

price, which for programs retailing for $7.95 will come to around $1 per cassette. On a sale of 50,000 this would amount to about $50,000 in royalties.

When you submit your program, be sure to check and recheck the cassette to make sure it loads properly on your own system. Send us a cassette, a listing of the program (if you have one), a write-up on how to use the program, who you think would be interested in it, what it does for the user, etc. The more complete the documentation, the better. If there are any variables that users might want to change, indicate them and the line numbers that should be changed.

If you are using other than an 80 or a PET, be sure to tell us what version of BASIC you are using, what cassette system of recording, etc.

Our Job

As mentioned, we'll check out your program with a panel of experts and try to come up with the best and most useful program possible. You'll be consulted on any changes. We will need a signed statement of original work from you to protect us in any copyright problems. If your work is not entirely original, you should get in touch with the programmer who did the original work and strike a deal with that person for the rights. Obviously, we prefer wholly original material . . . it is simpler to protect.

Once we are ready to go we'll duplicate your program and put it out on cassette . . . into the approximately 1000 computer stores, and, if it is for the 80, it could go into some 7500 or more Radio Shack stores. If it only sells an average of ten copies per store, you'll have 86,000 programs sold! Royalty checks will be coming every month.

Wiped Out!

One whole week down the drain! Well, it could have been worse, of course . . . it could have been a month. But perhaps you can imagine the shock when the *Kilobaud* Lab group discovered that the entire stack of software master cassettes had been zapped by accident.

With three of the Radio Shack TRS-80 systems being used to check out programs and make up cassette masters for duplication, the systems take up a good part of one workbench. When a cassette master was finished and checked out it was put in a stack of other masters. The only problem was that the stack of masters was right next to one of the TRS-80 power supplies. And do you know what they have in those power supplies? They have a transformer, complete with the usual magnetic field, and any tape put near it gets wiped out. Try it for yourself and see.

So it was back to the old drawing board . . . and a mad hunt through the many duplicated cassettes to reconstruct the master tapes. It only took a few days to get them back into shape, but with several thousand orders hanging fire even that time loss was painful. You may be sure that the power supplies are now well out of the way.

Contest!

The voting for the best article in the June issue was very heavy, and a preponderance of ballots were cast in favor of "8080, Z-80 or 8085" by Michael Slater.

Winner of a book from the KB Book Nook is Robert Herbold of the nation's capital.

To Michael and Robert we offer congratulations; to our readers, we say: Keep voting!

SYSTEM 710

COMPLETE BUSINESS COMPUTER ACCOUNTING SYSTEMS

WITH FULL PROFESSIONALLY DESIGNED SOFTWARE PACKAGES INCLUDING GENERAL LEDGER, ACCOUNTS RECEIVABLE, ACCOUNTS PAYABLE, POINT-OF-SALE INVENTORY, SORTING, MERGING AND MAIL LIST PROGRAMS. ALL PROGRAMS SHARE A COMMON DISK FILE DATA BASE AND THEREFORE, INSTANT, UP-TO-THE-MINUTE BALANCE SHEETS AND PROFIT/LOSS STATEMENTS ARE NO LONGER A DREAM, BUT ARE NOW REALITY.

DELTA SYSTEMS 700 & 710

SYSTEM 700

Disappointed by many of the "business" systems you've seen? Nice hardware, but no software, RIGHT? At last there is a totally integrated system (THAT WORKS!) available now which is just right for your needs and budget. Our SYSTEM 700 series of business computers offers completely interactive and self-prompting programs which even your secretary could operate after only 2-3 hours of familiarization.

THE SYSTEM ALSO OFFERS:

- Multi-User Capability (your parts room can run inventory while bookkeeping does general ledger)
- Word Processing (computer generated forms and letters which look individually typed)
- Broad Peripheral Base (6.4 Megabyte floppy disks, 29-70 Megabyte hard disks, Multi-Terminal Adapters, Modems, Controller Interfaces and more)
- Graphics Capability
- User Programmable (Compilers, Assemblers, etc.)
- Extensive Software Library including Cash Flow Analysis, Buy/Lease Analysis, Loan Analysis, Production/Man-hour Ratio Analysis programs, and many more Business/Management aids available at a nominal price.
- Full Factory and Dealer service and maintenance support nationwide.

6800 CPU

- Based on our *FAST*6800 Central Processing Unit which utilizes the industry standard S-100 mainframe for full compatability with many other S-100 products. (CPU available separately, call for prices)

STARTING AS LOW AS $5999*

*INCLUDING SOFTWARE GL, A/R, A/P, INVENTORY PROGRAM PACKAGE AB–1

DATATRONICS, INC.
208 EAST OLIVE · 303/336-7956
LAMAR, COLORADO 81052
Dealer and Distributor inquiries invited.

D30

See us at
BizComp '78
Booth 3
Atlanta, Ga.
OCT. 27-29

Kilobaud Classified

Kilobaud classified advertisements are intended for use by those individuals desiring to buy, sell, or trade used computer equipment or software. No commercial ads are accepted.

Two sizes of ads are available. The $5 box allows five lines of about 22 characters each, including spaces and punctuation. The $10 box provides ten lines of type – again, each line is about 22 characters. Minimize capital letters, as they use twice the space of small characters. Payment is required in advance with ad copy. We cannot bill, or accept credit. Oversize ads are not accepted. Each subscriber is limited to two (2) identical ads in any given issue.

Advertising text and payment must reach us 60 days in advance of publication. For example, advertising copy for the March issue (mailed in February) must be in our hands on January 1. The publisher reserves the right to refuse a questionable or not applicable advertisement. Mail advertisements to: KILOBAUD CLASSIFIED, Kilobaud, Peterborough, NH 03458. Do not include any other material with your ad, as it may be delayed.

Regulated power supplies, tested working 5 V @ 25 Amp, $95, freight collect, UPS. A. Khan, 2355 Burch Cir., N.E., Atlanta GA 30319. (404) 325-0089.

TRS-80 Level I 4K—5 pop. games. Orig.—graphics & score—cassette, $7. C. Zalnerunas, 3034 W. Columbus Ave., Chicago IL 60652.

Variety of programs on cassette for Radio Shack TRS-80. Details from Dr. S. Harter, Box 17222, Tampa FL 33682.

TRS-80, 4K min., Level I BASIC. Alpha/numeric graphics. Magnify letter/nums up to 6X. Excellent for displays. Displays more than 80 chars.; up to 16 chars. at a time; A-Z, 0-9 & some spec. chars. $10 on tape w/doc. To: EMFW, Box 438, Oakhurst NJ 07755.

Mod. 35 KSR, sprocket fd, 2 color ribbon, $500. Harding, 6724 Lynmont Dr., Charlotte NC 28212. (704) 364-4770 or 568-6856.

TRS-80, 4K min., Level 1 BASIC. Play Keno, Las Vegas style, with same odds & chances as a casino game. Play with 1 to 15 people at a time. Computer keeps track of winnings & nos played for each. Great family fun. $10 on tape w/odds & inst. To: EMFW, Box 438, Oakhurst NJ 07755.

Heath H8, 16K, H9, cassette, Ext. BASIC. Up & running. $1250. Also, 10 new DPI 8-in. disks, $25. I pay shipping. A. Thornburg, RR 2, Thompsonville IL 62890. (618) 627-2166.

North Star Software Specials: Investment Analysis Package: $25. Home/Family Package: $25. Game Pack: $15. All these disks are full and just packed with programs! Order now, or write for complete list. Herbert Schildt, 1007 N. Division, Urbana IL 61801.

TRS-80—Several business and accounting programs available on cassette. Depreciation, financial analysis, income tax, interest calculations, real estate selection and evaluation and more. Write: C. Jones, 658 Edinboro Rd., NW, Atlanta GA 30327.

Bally Owners and hopefuls are banding together for our mutual benefit to extract the maximum usefulness from this unit. Anyone interested is invited to write Bob Fabris, 3626 Morrie Dr., San Jose CA 95127 with a SASE.

TRS-80 Software w/doc. Level I BASIC—comes on cassette completely tested & debugged. *Games Package (7 games), $12. *Education Package, $12. *Stock or Option Analysis, $12. *Utility Analysis Pgm. (elec., gas, tel.) Analyze your outlay by month & yr., $10. All programs easily adapted to other BASICs. 15 yrs. experience assures quality. J. Fox, 27 Prince Wm. Rd., Morganville NJ 07751.

Sell: Computer terminal Hazeltine 2000 series, good condition, with detached keyboard, $475 plus shipping. Vic Jurkovic, 325 Wilson Ave., Westwood NJ 07675. (201) 664-6833.

TRS-80 Sci-Fi/Fantasy Games (4K, L1): Exciting & sophisticated simulations, which include: Galactic Blockade Runner, a space war game—$9.95. Magic Isle, a fantasy economic simulation—$7.95. Sci-Fi sampler, 3 games—$5.95. Space Colony—$7.95. TRS-80 Demo Program—$5.95. All include manuals. 20% off for 3 or more. SASE for game list. Tim Quinlan, 219 Washington Ave., Chelsea MA 02150.

We had replacement panels made for our Altair 8800s. They're blue against a white background and really look nice! We'll sell the extras for $16 each. D. L. Tibbetts, 520 Schoolhouse Ln., Willow Grove PA 19090.

IMP-16 Owners—Complete floating point package. Arithmetic, long shifts, DP integer and floating compare, I/O and 10 important functions—LN, EXP, X**Y, etc. Diskette or paper tape, operators manual and documentation. $100. T. Van Sickle, PO Box 556, Brighton MI 48116.

TRS-80 Electronic design software. Active filter design; low, high, and bandpass, Level-I, 4K cassette, $9.50. Other tapes available, list $1. J and J Hansen, 6890 50th St., San Diego CA 92120.

For sale: Heath H8, 16K, H9, cassette, up & run, $1200. David Marcus 430 Wolf Hill Rd., Dix Hills NY 11746. (516) 427-1926.

TRS-80 OWNERS!! Original software including business system for Amway Distributors. Level I & II versions available. All will run in 4K. Also have moving signboard machine-language program. Write for info. Allan E. Sitter, 8 Driftwood Circle, Groton CT 06340.

Amortization for TI 59 with PC100a printer. Good annotation. Includes Pmt. #, interest, principal, new bal. Send $5 plus magnetic card. George, PO Box 1576, Bakersfield CA 93302.

TRS-80, PET & Apple II reference list of software on cassettes. Published in Aug. and every 3 months. Cost is $1 to North America and $2 elsewhere. Robert Purser, Box 466, El Dorado CA 95623.

Programs For: SOL, SOL BASIC 8, North Star BASIC, SOL/North Star. Super DOS, Personalized, DOS for SOL, $5. Bomb, Suspense Game, $4. Star-Ship Trainer, $4. Convert, $2.50. Octal/Hex Loader, $3. Plus Many More!! Above are listings. For cassette add $2.50, Mini-disk, $5.25. Send for complete list to: Pete Pacione, 2952 N. Meade, Chicago IL 60634.

Motorola D2 Users: A manual of assembly-language problems. Step-by-step approach of 6800 instructions, digital input/output, interrupts and many programming techniques, $5. K. Rao, Department of Physics, Western Michigan University, Kalamazoo MI 49008.

TRS-80 Monthly Newsletter. For information write to: Howard Y. Gosman, Box 149, New City NY 10956.

*Books of computer games in BASIC. ENIGMAS-1 ($8): Gone Fishing, Concentration, Starship, Craps, Slot-Machine, Sherlock Holmes, Tank Attack. ENIGMAS-2 ($8): Number Guess, Mortar Battle, In-Between, Shell Game, Safari, Starship-2, Dice Roll, Puzzle. *Catalog and test program Frog Race, $.50. Available in Standard BASIC, SWTP 8K BASIC or Radio Shack TRS-80 BASIC. Please specify. B. Erickson, PO Box 11099, Chicago IL 60611.

TRS-80 Software: Business programs written in 16K Level II. Available for cassette or disk files. Send for list & prices. L. Owens, Rt. 6, Box 336A, Thomasville GA 31792.

For sale: Apple-II Software-cassette: Author-Title Index program for books, records, tapes, super-fast mach. lang. sort, 16K Blackjack, graphic, paddle input, sound, full L. V. rules, auto-play by computer, $10 each incl. doc. Both for $15. George W. Lee, 18803 S. Christina Ave., Cerritos CA 90701.

PET Owners: A craps program loaded with features like animated dice, lowercase literals, statistical analysis with bar graph and much more. 265 lines, 5K bytes; on tape and ready to roll. Send $7.50 to Richard Liebert, PO Box 268, Scarborough Station NY 10510.

TRS-80 machine language routines to write cassettes which will load using CLOAD, T-BUG, RSM-1, or other monitors. Complete listings and instructions. Send $3.95 to Gene Perkins, 5224 Winifred, Ft. Worth TX 76133.

PET Owners: Send for free catalog of PET software. Custom programming available. C. M. Stuart, 5115 Menefee Dr., Dallas TX 75227.

For sale: Heathkit Computer System—H8 computer, H8-1 memory board (4K), H8-3 chip set (4K), H8-5 serial I/O & cassette interface (1 or 2 recorders), H8-2 parallel interface, ECP-3801 cassette recorder, H9 video terminal. Assembled and working; asking $1500. Thomas C. Smith, 506 19th Lane, Vero Beach FL 32960. (305) 569-4622 after 6 PM

Heath: cassettes—Listings—Time—Trek—Day—Convert Nos—Backgammon. BASIC LIBRARY. (401) 331-5034 or (401) 738-3265.

TRS-80 Music generation program; no hardware needed. Cassette (Level I and II) and instructions, $9.95. Write: D. Palmer, 815 Ky Ave., Lynn Haven FL 32444.

TRS-80 Level II, 16K. Star Trek game that runs in Level II, $5.95. Other games SASE. J. Joyce, 5925 Rockhill Rd., Kansas City MO 64110.

All prime, 100% guaranteed, in tubes. CD4030—25/$7.50, CA3079—25/$12, 74S112—25/$10, 74116—15/$15. A. Khan, 2355 Burch Cir. N.E., Atlanta GA 30319.

TRS-80 software for small-fries. Flash Cards, a math game for kids 8 yrs. and older. Hi-Lo for 2, for kids who can count to 100. $4 ea. or both on same cassette for $7. Jim Creasy, 517 Reeves Dr., Phoenixville PA 19460.

TRS-80 software on cassette or diskette. Lowest cost. Free newsletter. Poi Pow, 96 Dothan St., Arlington MA 02174.

PET users need info? Getting started with your PET workbook, $4. Workbook has helpful info, sample programs, exercises and fundamental features of PET BASIC. Send SASE for details. D. Smith, Box 921, Los Alamos NM 87544.

SWTP new AC-30 cassette interface, assembled, documented: $75. GT-6144 graphic system, assembled, documented: $79. MP-L parallel interface, assembled: $25 with GT-6144. D. O'Hair, Box 17-8135, San Diego CA 92117.

CIRCLE CHESS instruction book by Alphonso, used in Circle Chess software development. $5, Stanonis, Box 63, Des Plaines IL 60017.

TRS-80 Programming Contest—Win $500—Send self-addressed stamped envelope to: TRS-80 P.C.-K, PO Box 621, Fenton MO 63026.

Use the Classifieds

KB CALENDAR

ComputerLand

ComputerLand Corp. will celebrate the shared success of its stores, customers and friends during a 2nd year birthday, Sept. 3 and 23 at all ComputerLand locations (growing at 1 a week, with operations recently expanded into Europe). Everyone is invited.

Washington DC

COMPCON FALL '78, sponsored by IEEE Computer Society, is being held in Washington DC, September 5-8, 1978, at the Capital Hilton. For a copy of the advance program, write to: COMPCON FALL '78, PO Box 639-g, Silver Spring MD 20901.

Dallas TX

International Microcomputer Exposition, Dallas—Sept. 29-30-Oct. 1, 1978. Dallas Convention Center, 413 Carillon Tower, 3601 Preston Rd., Dallas 75240, (214) 271-9311.

Boston MA

The Boston Computer Society presents "Home/Business Computers '78 . . . An exposition of the state of the art in microcomputers," at Sherman Union, Boston University, 775 Commonwealth Ave., 10 AM to 5 PM on October 7, 1978. For more information call 864-7291 or write: The Boston Computer Society, 17 Chestnut Street, Boston MA 02108.

WIREWRAP

PRECUT WIRE

Why buy wire on rolls?

PRECUT & STRIPPED WIRE IS:

- **Fast** - No more cutting & stripping by hand
- **Reliable** - Good, clean, uniform strip
- **Economical** - Cheaper than using bulk wire

Precut Wire	Bulk Wire
100 pcs of 2" at $.82 = 3¼¢/ft.	50 ft. roll at $1.99 = 4¢/ft.
100 pcs of 6" at 1.06 = 2¢/ft.	100 ft. roll at 2.95 = 3¢/ft.
Wire Kit 1 at $6.95 = 2 1/3¢/ft.	

30 Kynar stripped 1" on each end. Lengths are overall. Colors: Red,Blue,Green,Yellow,Black,Orange,White. Wire packaged in plastic bags. Add 25¢/length for tubes.

	100	500	1000	5000
2½ in.	.78	2.40	4.30/K	3.89/K
3 in.	.82	2.60	4.71/K	4.22/K
3½ in.	.86	2.80	5.12/K	4.55/K
4 in.	.90	3.00	5.52/K	4.88/K
4½ in.	.94	3.21	5.93/K	5.21/K
5 in.	.98	3.42	6.34/K	5.52/K
5½ in.	1.02	3.65	6.75/K	5.86/K
6 in.	1.06	3.85	7.16/K	6.19/K
6½ in.	1.15	4.05	7.57/K	6.52/K
7 in.	1.20	4.25	7.98/K	6.85/K
7½ in.	1.25	4.45	8.39/K	7.18/K
8 in.	1.29	4.65	8.80/K	7.53/K
8½ in.	1.32	4.85	9.21/K	7.86/K
9 in.	1.36	5.05	9.62/K	8.17/K
9½ in.	1.40	5.25	10.03/K	8.50/K
10 in.	1.45	5.51	10.44/K	8.83/K
Add'l in.	.10	.41	.82/K	.66/K

WIRE KITS

#1 $6.95 **#2 $19.95**

#1			
250	3"	100	4½"
250	3½"	100	5"
100	4"	100	6"

#2					
250	2½"	250	4½"	250	6"
500	3"	250	5"	100	6½"
500	3½"	100	5½"	100	7"
500	4"	1 250 ft. Roll Bulk			

Choose One Color or Assortment

WIRE WRAP SOCKETS

	1-9	10-24	25-99	100-249	250-999	1K-5K
8 pin*	.41	.38	.35	.31	.29	.27
14 pin*	.36	.34	.32	.30	.28	.27
16 pin*	.38	.36	.34	.32	.30	.29
18 pin*	.63	.58	.54	.47	.44	.41
20 pin	.84	.78	.71	.63	.59	.54
22 pin*	.95	.90	.85	.75	.72	.70
24 pin	.91	.84	.78	.66	.64	.59
28 pin	1.25	1.15	1.08	.95	.89	.82
40 pin	1.65	1.55	1.42	1.25	1.15	1.09

Gold 3-level Closed Entry Sockets

* End & Side Stackable   All prices include gold

2-level sockets available

WIRE WRAP TOOLS

$34.95

HOBBY WRAP
Model BW 630

With Free Wire Kit 1
($6.95 Value)

Batteries & Charger	$11.00
WSU 30 Hand Wrap-Unwrap Strip Tool	5.95
WSU 30M, for Modified Wrap	6.95
BT 30 Extra Bit	2.95

INTERCONNECT CABLES

Ribbon cable connectors for connecting boards to front panels, or board to board

	SINGLE ENDED			DOUBLE ENDED		
	14 pin	16 pin	24 pin	14 pin	16 pin	24 pin
6"	1.24	1.34	2.05	2.24	2.45	3.37
12"	1.33	1.44	2.24	2.33	2.55	3.92
24"	1.52	1.65	2.63	2.52	2.78	4.31
48"	1.91	2.06	3.40	2.91	3.17	5.08

PAGE DIGITAL ELECTRONICS P36
135 E. Chestnut St. Suite #5
Monrovia, California 91016
Phone (213) 357-5005

Ordering Information:

- Orders under $25 and COD's, add $2
- All others, shipped Ppd in U.S. via UPS
- For Blue Label (Air) or 1st Class, add $1
- We accept Visa & Mastercharge
- Most orders shipped same day

Dealer Inquiries Invited

NEW SOFTWARE AVAILABLE FOR

NORTH STAR ★ COMPUTERS

The following software is now being offered for use on the North Star disk systems and Horizon Computers.

CP/M™ FDOS and Utilities		From $145
Microsoft FORTRAN-80		$400
Microsoft COBOL-80		$625
Microsoft Disk Extended BASIC		$300
Xitan SUPER BASIC	(A3)	$99
Xitan DISK BASIC	(A3+)	$159
Xitan Z-TEL Text Editor	(A3, A3+)	$69
Xitan Text Output Processor	(A3, A3+)	N/A
Xitan Macro ASSEMBLER	(A3, A3+)	$69
Xitan Z-BUG	(A3+)	$89
Xitan LINKER	(A3+)	$69
Xitan Package A3 (as keyed above)		$249
Xitan Package A3+ (as keyed above)		$409
Xitan Fortran IV		$349
CBASIC Compiler/Interpreter BASIC		$95
MAC Macro Assembler		$100
SID Symbolic Instruction Debugger		$85
TEX Text Formatter		$85
BASIC-E Compiler/Interpreter BASIC		$30
Accounts Receivable		$750
NAD Name & Address Processor		$79
QSORT Disk File Sort/Merge Utility		$95

Available from computer stores nationwide or order direct from:

LIFEBOAT ASSOCIATES
164 W. 83rd Street
New York, N.Y. 10024
(212) 580-0082 L16

8K BASIC for KIM-1

The famous high speed *Microsoft* BASIC is now available for KIM-1. The 9-digit accuracy version is available on cassette or paper tape.

DELIVERY FROM STOCK.

$99.00

$2.00 shipping and handling

JOHNSON COMPUTER J4
P.O. BOX 523
MEDINA, OHIO 44256 (216) 725-4560

SPACE AGE COMPUTER STORE

NEW YORK CITY
and
LONG ISLAND

IF YOU WANT PROFESSIONAL SERVICE IN A CASUAL ATMOSPHERE . . . AND A LARGE VARIETY OF EQUIPMENT

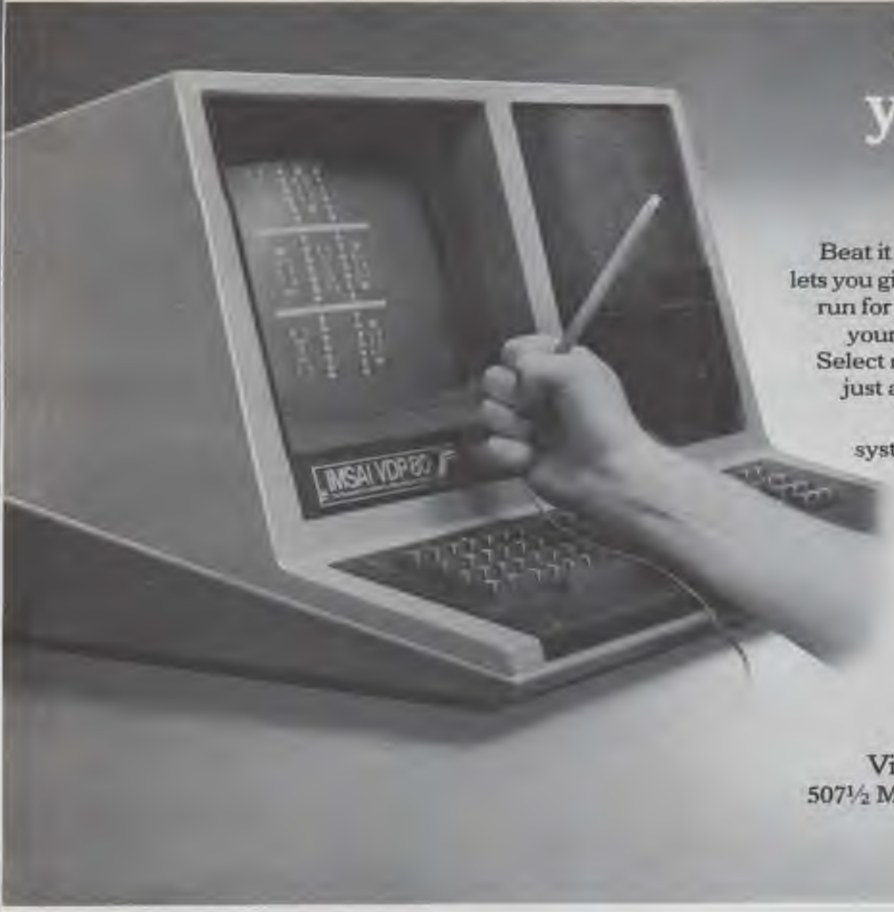
- **BUSINESS APPLICATIONS**
  General Ledger, Inventory, Accounts Payable, Accounts Receivable, Word Processing Systems.
- **GENERAL APPLICATIONS**
  Northstar Macro Assembler $65
  Northstar Mailing Label Program $45
- **STOCK MARKET PACKAGE** (Unique) Makes Makes Ticker-Tape Obsolete.
  Send $2.00 for descriptive brochure on Stock Market Package.

BYTE SHOP B14
THE AFFORDABLE COMPUTER STORE T.M.

130 East 40th Street	2721 Hempstead Turnpike
New York, NY 10016	Levittown, NY 11756
(212) 889-4204	(516) 731-8116
(Corner Lexington Avenue)	(Just East of Wantagh Pkwy)

Tues. thru Fri. 11 to 7 . . . . 12 to 8
Saturday 10 to 5 . . . . 10 to 5

Start beating your computer.



Beat it with a Vidiet-Stik, a little light pen that lets you give any Z80 or 8080 based system a real run for its money. Vidiet-Stik piggybacks with your keyboard allowing you to play games. Select menus. Set up educational drills. Or do just about anything else you feel like doing.

The pen is easily integrated into your system, requiring only +5v, ground, and a single input bit. And each fully assembled and tested Vidiet-Stik comes with complete interface instructions as well as documentation including driver, test and game software.

The price is a very light $39.95 plus $1.50 for postage and handling. (Indiana residents add 4% sales tax.)

Vidiet-Stik. from **Esmark*** **Inc.**
507½ McKinley Hwy. Mishawaka IN 46544

*Electronic Systems Marketing

$199.95
AS IS

$249.95
WORKING & CLEANED

IBM® Selectric-Based I/O Writers Excellent Hobby Printers

Series 72/731	All Solenoids
Heavy Duty	Original Documentation
8½" Platten	While Supply Lasts

SUPER SALE

These terminals are from a large airline reservation system. They are heavy duty and were under continuous maintenance. The units have been in storage. We make every effort to ensure that all essential parts are included. Most work when plugged in. No warranties are given or implied.

Conversion Kits.

1. Conversion instructons, P C board for printer only using software approach $59.95
2. I/O kit makes the unit into a conversational terminal. Instructions, P C board, components for a parallel or RS232 interface. Will work with any IBM terminal $249.95
3. Completely converted unit and assembled interface for I/O using kit $999.95

Card reader by HP with RS232 interface	$299.95
Printec line printers, parallel interface	$1500.00
Honeywell 516 & 316 mini's, make offer. Cables, used 11 conductor, 100 ft with connectors	$9.99
Electronic parts and circuit boards 1/4 lb bag	$4.99

Check, Money Order, Cash. Personal checks require 3 weeks to clear. No COD's. Units shipped UPS or PP collect. Prices Net FOB Tulsa

SUPER SURPLUS SALES S55
P.O. BOX 45944 TULSA, OK 74145 1-918-622-1058

Don Lancaster's ingenius design provides software controllable options including:

- **Scrolling • Full performance cursor**
- **Over 2K on-screen characters with only 3MHz bandwidth**
- **Variety of line/character formats including 16/32, 16/64 .... even 32/64**
- **User selectable line lengths**

TELL ME MORE! ( ) Send instruction manual for the TVT-6 Kit with full operational details. $1 enclosed.
( ) SEND FREE CATALOG

P9 Name: ______

PAiA Address: ______

ELECTRONICS, INC. City: ______ State: ______ Zip: ______

DEPT. 6–K, 1020 W. WILSHIRE BLVD., OKLAHOMA CITY, OK 73116

TRS-80

SOFTWARE EXCHANGE

JOIN IN ON THE FUN!!!!
THE TRS-80 SOFTWARE EXCHANGE CIRCUIT HAS FORMED

HOW TO JOIN

Send (on cassette) your best general interest program. In return, we send you one of our Circuit Exchange Cassettes (6-10 programs per tape). Programs meeting our standards are considered for retail sales, and you receive commissions on every program sold.

NOTE: Looking for quality material for projected newsletter. Interested parties contact:

Roger Robitalle
Box 61
Milford, New Hampshire
03055

Write: TRS-80 SOFTWARE EXCHANGE
17 Briar Cliff Drive
Milford, N. H. 03055 T38

Program faster, with fewer errors

Save time, save money, have more fun when you write programs in BASIC on our new combination Program Coding/CRT Layout Sheets.

Gives you 2 form uses for the price of one. Use full 28 line x 80 column grid area for regular program steps. To plan display layouts, simply keep your characters within the appropriate CRT indicator lines, and you'll automatically see where they'll be on your CRT screen.

Gives you 4 extra coding lines on every sheet. Full 28 line x 80 column coding capacity saves you 14 sheets out of every 100, compared to 24-line forms, yet we offer full-size 6mm x 3mm grid blocks.

Works with your CRT display, no matter what brand you own. Equipped for both 16 line x 64 column and 24 line x 80 column display formats.

Surface-engineered to take both pen & pencil without blotching. Gives you crisp, sharp characters whether you write with pencil or plastic-tip pens.

Eye-comfortable soft blue grid. All grid rulings, tints, and division rules are easy on your eyes, even after hours of continuous programming.

Bright white sheet makes your characters easy to read. Heavy 22-pound brilliant white opaque paper gives your writing contrast.

8½ x 11 size saves you money. . .fits standard binders, folders, & files. Designed to serve professional programmers, hobbyists, and small business people economically.

Your choice of looseleaf style or 50-sheet pads. Both are 3-hole-punched. The choice is yours.

Try a pack today. Ask your local computer store for Stirling/Bekdorf 78C1's. Or use the coupon below to get them right now.

YES. Rush me the quantity of 78C1 BASIC Coding/CRT Layout Sheets I have indicated below:

Looseleaf	Pads
☐ 100 – $6.05 + $1.50 shpg	☐ two 50-sht. pads: $6.35 + $1.50 shpg
☐ 500 – $24.10 + $2.85 shpg	☐ ten 50-sht. pads: $26.85 + $2.85 shpg

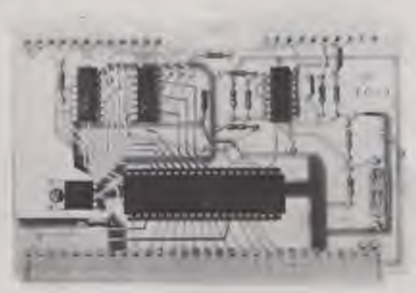
☐ Enclosed is my check for $ ______
☐ Charge to: ☐ Master Charge; ☐ Visa Card # ______

Name ______
Address ______
City ______ State ______ Zip ______

Stirling/Bekdorf™ S54

K9 4407 Parkwood □ San Antonio, TX 78218 □ (512) 824-5643

High Performance Cassette Interface



For SWTPC Computers

- **FAST** - 4800 Baud Loads 4K in 8 Seconds!
- **RELIABLE** - Error Rate Less Than 1 in 10^6 BYTES.
- **CONVENIENT** - Plugs Directly Into The **SWTPC** Motherboard.
- **LOW COST** - $49.95 For Complete Kit.
- **PLUS** - A Fully Buffered 8 Bit Output Port Is Also Provided.

The TC-3 Cassette Interface Records In A Modified FM Format Similar To Disk Systems. It Is Edge Sensitive And Immune To Phase Reversals In The Recorder.

"Designed By Professionals For Outstanding Performance"

JPC PRODUCTS CO.
P.O. BOX 5615
ALBUQUERQUE, N.M. 87185

J12

Terms: Cash or MC
Shipped Prepaid
NM Residents Add 4% Tax.

S-100 MAINFRAME $200

COMPUTER SHOPPING?

The computer Data Directory can save you hours of time in locating computer products and company addresses. Contains Profiles of over 675 small computer manufacturers, stores, distributors, publishers, and electronic dealers. All addresses are up-to-date and at your finger tips! Over 100 pages. Includes reader service card to all firms listed. Definitely a useful reference source. Order yours today

ORDER TODAY
Only $4.98
Plus $1.00 Postage and Handling

SATISFACTION GUARANTEED
OR YOUR MONEY BACK

THE *Computer Data Directory* C77

"The Index of Small Computer Products and Companies"

Box. 598 Cleveland, Ohio 44107

Please send me ______ copy(ies)

Name ______________________
Street ______________________
City ______________ State ______________ Zip ______________
Enclose $4.98 Each (Plus one dollar postage and handling)

commodore **Radio Shack**

PET TRS-80

EITHER WAY... We've got software for you!

You can find out what our customers already know—Personal Software consistently offers great software products. Check out the programs below—they each represent many man-months of expert programming effort. We're sure you'll be pleased with the results.

6502 ASSEMBLER IN BASIC by **Dan Fylstra** for 8K PETs: Accepts all standard 6502 instruction mnemonics, pseudo-ops and addressing modes. Evaluates binary, octal, hex, decimal, and character constants, symbols and expressions. Assembles object programs anywhere in memory. Includes one and two pass versions of the assembler, text editor and disassembler, with a 30 page manual and PET machine language programming hints **$24.95**

MICROCHESS 1.5 by **Peter Jennings** for 4K Level I and II TRS-80s: In Z-80 machine language, easily loaded from cassette using the CLOAD command (TBUG is not needed). Uses standard algebraic chess notation to describe moves, and checks every move for legality. Handles castling and en passant captures. You can play white or black, set up and play from special board situations, or even watch the computer play against itself! With 3 levels of chess play **$19.95**

BRIDGE CHALLENGER by **George Duisman** for 8K PETs and 16K Level II TRS-80s: You and the dummy play four person Contract Bridge against the computer. The computer will deal hands at random or according to your criterion for high card points. You can review tricks, swap sides or replay hands when the cards are known. No longer do you need four people to play! **$14.95**

ORDERS: *Check, money order or VISA/Master Charge accepted; programs and cassettes guaranteed. Our catalog describes many other great software products, including an ASTROLOGY program, a FOOTBALL game, a GRAPHICS utility package and many others. For your free copy, send a letter giving your PET or TRS-80 serial number, memory size, and your most wanted software product.*

VISA **Personal Software™** master charge
P.O. Box 136-K9, Cambridge, MA 02138
VISA/MC telephone orders welcome at (617) 783-0694

JOHNSON COMPUTER J4
P. O. BOX 523, MEDINA, OHIO 44256
(216) 725-4560

FINALLY!

HDE inc. FLEXIBLE DISK SYSTEM FOR KIM

Features:
- *Top Quality Industrial Grade Controller
- *Proven High Reliability Disc Drive
- *Line-Numbered Text Entry and Editing
- *A Powerful Command Structure
- *Adaptation to any 6500 Based System
- *Capability for User Defined Commands
- *Complete Compatibility with KIM
- *Multiple Resident Files
- *Indexed and Non-Indexed Disk Storage

Complete 90 Day Parts and Labor Warranty

HDE FILE ORIENTATED DISK SYSTEM
"FODS"

Includes:
- *Full Size Sykes Drive
- *6502 Based Controller
- *Power Supply
- *FODS Software
- *Cables, Interface Card
- *User Manual

COMPARE AND SAVE

	LIST PRICE	OUR CASH PRICE
IMSAI 8080 Kit	699^{00}	569^{95}
North Star MDS A Kit	699^{00}	589^{95}
with Power Supply, plus Cabinet and 3 Extra Diskettes	790^{50}	659^{95}
Horizon I Kit	1599^{00}	1339^{00}

Check Our Prices on:
Cromemco, Xitan, IMSAI, Vector Graphic, Sanyo, Hazeltine plus most other major lines.

Above Prices Include Cash Discounts.
Shipping Charges Extra.
Subject To Available Quantities.

Bus. . . . S—100, inc. S60
Address . . . **7 White Place**
Clark, N.J. 07066
Interface . . . **(201) 382-1318**

Bits Bytes Books Bargains

IMSAI et al

MARKETLINE SYSTEMS, Inc.

2337 Philmont Ave. M17
Huntingdon Valley, Pa. 19006
215/947-6670 · 800/523-5355

D.C. METRO AREA
TYSONS CORNER, VIRGINIA

THE COMPUTER SYSTEMS STORE

MICROCOMPUTERS & PERIPHERALS
HOME, SCHOOL & SMALL BUSINESS

FEATURING

Small Business Systems Complete with Application Software Including Word Processing

DEALERS FOR

Commodore Pet	Lear Siegler
Processor Technology	Diablo
Polymorphic	Texas Instruments
Cromemco	DEC
Southwest Technical	North Star
Micro-Computer Business Systems, Inc.	Anderson Jacobson Books & Magazines

CSS

1984 Chain Bridge Rd
McLean, VA
703-821-8333 C73

TRS-80 16K RAM CONVERSION KIT

SAVE OVER $125.00!

This easy-to-use kit contains 8 dynamic memory chips UPD416 (1 X 16K) and instructions on converting your 4K TRS-80 to 16K.

ONLY $165.00!

CALL YOUR ORDER IN
(404) 458-4690

VISA

DELTRONIKS

PO BOX 29363 D28
ATLANTA, GA 30359

SELECTOR II. . .

- **AN ON-LINE REPORT GENERATOR**
- **AN ON-LINE FILE MANAGER**
- **REQUIRES NO PROGRAMMING EXPERIENCE**
- **FOR CP/M / MICROSOFT EXTENDED DISK BASIC USERS**

SELECTOR II extracts selected information from your files according to your requirements, sorts it, and prints it in a formatted report.

Permanent license price of $225 includes 1 year maintenance. Supplied on 8'' single-density diskette, with user's manual.

Master Charge or Visa accepted.
MICRO-AP M51
8939 San Ramon Road
Dublin, CA 94566

APPLE OWNERS

Interactive **Trap and Chase** games for two people. Software allows choice of speed and points and requires 8K RAM. Hardware consists of two game control boxes. Each box has four micro switches and specially designed printed circuit with 5 feet of cable completely assembled. Plugs right into your game I/O connector. Control boxes can also be used with the game Dragon Maze

Hardware and software cassette—$49.95. (Texas residents add 5% sales tax.) 90 day warranty parts and labor. Guaranteed 30 day delivery. Allow time for processing personal checks. Send check or money order to:

B & G Interfaces
P.O. Box 59364 B28
Northhaven Sta.
Dallas, Tx. 75229

UNUSUAL SOFTWARE FOR TRS-80!

$10.00 each, on cassette

UTILITARIAN PROGRAMS

CS-7 **LEND OUT**—A program jam-packed with automatic and semi-automatic features! For keeping track of items you lend to other people. Automatically assigns to a loaned item an identification number and files this number on tape along with such information as: what the item is, who borrowed it, and the date it was borrowed. Program updates file when item is returned without affecting the rest of the file. Plus even more features!

CS-6 **ITINERARY**—Prepares a personal itinerary. Stores memos with their appropriate times in chronological order. Will recall itinerary as a whole or in parts. Even after an itinerary is filed, a new memo can be inserted into it between any two previously filed memos. More!

SCIENTIFIC CURIOSITIES

CS-1 **BIOFORECAST**—More advanced than biorhythm. Uses more exact non-sinusoidal functions. Gives not only the three cycles, but also their interactions. More! With special interpretation chart.

CS-2 **ORBIT**—4 programs figure data on orbits around the earth and sun.

CS-3 **INTERSTELLAR**—Predicts facts concerning possible voyages to more than 30 different stars at relativistic speeds.

*Add $1.00 postage & handling. All programs Level-1.

Computrex C86
PO Box 536 Inman SC 29349

SURPLUS ELECTRONICS

ASCII ASCII

IBM SELECTRIC® BASED I/O TERMINAL WITH ASCII CONVERSION INSTALLED $695.00

- Tape Drives • Cable
- Cassette Drives • Wire
- Power Supplies 12V15A, 12V25A, 5V35A Others, • Displays
- Cabinets • XFMRS • Heat Sinks • Printers • Components

Many other items
Write for free catalog
WORLDWIDE ELECT. INC. W16
10 FLAGSTONE DRIVE
HUDSON, N.H. 03051
Phone orders accepted using VISA or MC. Toll Free 1-800-258-1036
In N.H. 603-885-3705

AD-68A ANALOG-TO-DIGITAL CONVERTER

- **8 analog input channels**
- **Occupies one SWTPC 6800 I/O slot**
- **Quality components and PCB**
- **Complete documentation**
- **90 day full warranty**
- **Assembled and tested**

$39.95

Available from your local dealer or factory direct. Write for additional information.

INNOVATIVE TECHNOLOGY
I24
510 Oxford Park
Garland, Texas 75043
(214) 270-8383

FIRST CLASS
Permit No. 17
Peterborough NH 03458

BUSINESS REPLY MAIL

No postage stamp necessary if mailed in the United States

Postage will be paid by

kilobaud

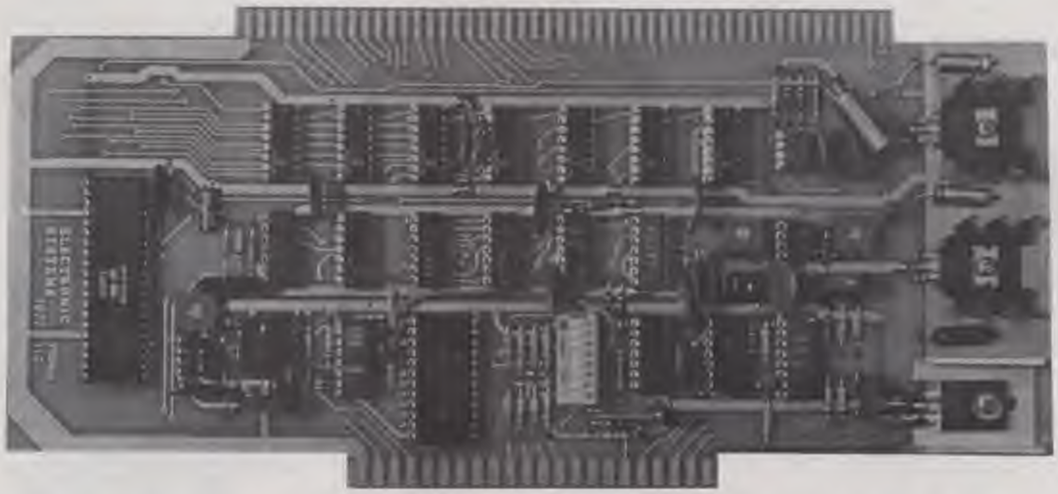
Peterborough NH 03458

COMPUTER INTERFACES & PERIPHERALS

SAVE

For free catalog including parts lists and schematics, send a self-addressed stamped envelope.

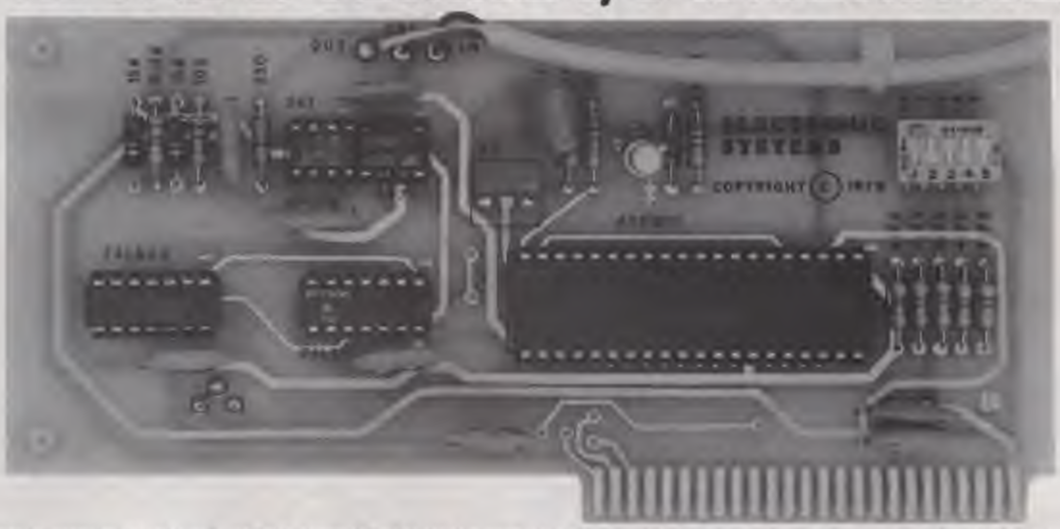
TIDMA



With this Tape Interface Direct Memory Access board you can record and play programs without a bootstrap loader (no prom). This can be up and running as fast as putting it in the socket. It takes complete control of the S-100 bus and reads or writes directly into the memory without the need of the microprocessor. It has an FSK encoder/decoder for direct connections to a low cost cassette recorder at 1200 baud, and inputs and outputs for direct connections to a digital recorder at any baud rate. This board is S-100 bus compatible.

Board only, $35.00 Part No. 112; with parts $110.00 Part No. 112A.

APPLE II SERIAL I/O INTERFACE



This serial I/O port is continuously adjustable from 0 to 30,000 baud and plugs into any peripheral connector. Features a low current drain, RS-232 input and output, and jumper selectable address. On board switch is selectable 5 to 8 data bits, 1 or 2 stop bits, and parity or no parity either odd or even. SOFTWARE: input and output routine from monitor or Basic to teletype or other serial printer; program for using an Apple II for a video or an intelligent terminal; can output in correspondence code to interface with some selectrics.

Board only, $15.00 Part No. 2; with parts, $42.00 Part No. 2A; Assembled and tested, $62.00 Part No. 2C.

ELECTRONIC SYSTEMS

Dept. KB, P.O. Box 21638, San Jose, CA. USA 95157

To Order: Mention part number and description. For parts kits add "A" to part number. In USA, shipping paid for orders accompanied by check, money order, or Master Charge, BankAmericard, or VISA number, expiration date and signature. Shipping charges added to C.O.D. orders. California residents add 6.5% for tax. Outside USA add 10% for air mail postage, no C.O.D.'s. Checks and money orders must be payable in US dollars. Parts kits include sockets for all ICs, components, and circuit board. Documentation is included with all products. All items are in stock, and will be shipped the day order is received via first class mail. Prices are in US dollars. No open accounts. To eliminate tariff in Canada boxes are marked "Computer Parts." Dealer inquiries invited.

24 Hour Order Line (408) 226-4064

✱ Circuits designed by John Bell

S A V E

COMPUTER INTERFACES & PERIPHERALS

S A V E

For free catalog including parts lists and schematics, send a self-addressed stamped envelope.

APPLE II SERIAL I/O INTERFACE *

art no. 2
aud rate is continuously adjustable rom 0 to 30,000 • Plugs into any periph- ral connector • Low current drain. RS- 32 input and output • On board switch electable 5 to 8 data bits, 1 or 2 stop its, and parity or no parity either odd or ven • Jumper selectable address • SOFTWARE • Input and Output routine rom monitor or BASIC to teletype or other serial printer. Program for using an Apple II for a video or an intelli- ent terminal. Also can output in correspondence code o interface with some selectrics. Board only — $15.00; with parts — $42.00; assembled and tested — $62.00.

T.V. TYPEWRITER

Part no. 106
• Stand alone TVT • 32 char/line, 16 lines, modifications for 64 char/line included • Parallel ASCII (TTL) input • Video output • 1K on board memory • Output for computer controlled curser • Auto scroll • Non-destructive curser • Curser inputs: up, down, left, right, home, EOL, EOS • Scroll up, down • Requires +5 volts at 1.5 amps, and -12 volts at 30 mA • All 7400, TTL chips • Char. gen. 2513 • Upper case only • Board only $39.00; with parts $145.00

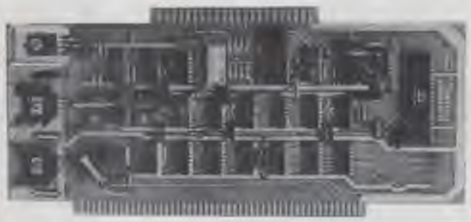
8K STATIC RAM

Part no. 300
• 8K Altair bus memory • Uses 2102 Static memory chips • Memory protect • Gold contacts • Wait states • On board regulator • S-100 bus compatible • Vector input option • TRI state buffered • Board only $22.50; with parts $160.00

MODEM *

Part no. 109
• Type 103 • Full or half duplex • Works up to 300 baud • Originate or Answer • No coils, only low cost components • TTL input and output-serial • Connect 8 ohm speaker and crystal mic. directly to board • Uses XR FSK demodulator • Requires +5 volts • Board $7.60; with parts $27.50

TIDMA *



Part no. 112
• Tape Interface Direct Memory Access • Record and play programs without bootstrap loader (no prom) has FSK encoder/decoder for direct connections to low cost recorder at 1200 baud rate, and direct connections for inputs and outputs to a digital recorder at any baud rate. • S-100 bus compatible • Board only $35.00; with parts $110.00

RF MODULATOR *

Part no. 107
• Converts video to AM modulated RF, Channels 2 or 3. So powerful almost no tuning is required. On board regulated power supply makes this extremely stable. Rated very highly in Doctor Dobbs' Journal. Recommended by Apple. • Power required is 12 volts AC C.T., or +5 volts DC • Board $7.60; with parts $13.50

DC POWER SUPPLY *

art no. 6085
Board supplies a regulated +5 volts t 3 amps., +12, -12, and -5 volts at amp. • Power required is 8 volts AC t 3 amps., and 24 volts AC C.T. at 1.5 mps. • Board only $12.50; with arts excluding transformers $42.50

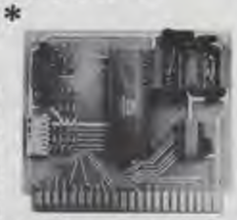
RS 232/TTY * INTERFACE

Part no. 600
• Converts RS-232 to 20mA current loop, and 20mA current loop to RS-232 • Two separate circuits • Requires +12 and -12 volts • Board only $4.50, with parts $7.00

TAPE INTERFACE *

Part no. 111
• Play and record Kansas City Standard tapes • Converts a low cost tape recorder to a digital recorder • Works up to 1200 baud • Digital in and out are TTL-serial • Output of board connects to mic. in of recorder • Earphone of recorder connects to input on board • No coils • Requires +5 volts, low power drain • Board $7.60; with parts $27.50

UART & BAUD RATE GENERATOR *



Part no. 101
• Converts serial to parallel and parallel to serial • Low cost on board baud rate generator • Baud rates: 110, 150, 300, 600, 1200, and 2400 • Low power drain +5 volts and -12 volts required • TTL compatible • All characters contain a start bit, 5 to 8 data bits, 1 or 2 stop bits, and either odd or even parity. • All connections go to a 44 pin gold plated edge connector • Board only $12.00; with parts $35.00 with connector add $3.00

RS 232/TTL * INTERFACE

Part no. 232
• Converts TTL to RS-232, and converts RS-232 to TTL • Two separate circuits • Requires -12 and +12 volts • All connections go to a 10 pin gold plated edge connector • Board only $4.50; with parts $7.00 with connector add $2.00

ELECTRONIC SYSTEMS

Dept. KB, P.O. Box 21638, San Jose, CA. USA 95157

To Order:

Mention part number and description. For parts kits add "A" to part number. In USA, shipping paid for orders accompanied by check, money order, or Master Charge, BankAmericard, or VISA number, expiration date and signature. Shipping charges added to C.O.D. orders. California residents add 6.5% for tax. Outside USA add 10% for air mail postage, no C.O.D.'s. Checks and money orders must be payable in US dollars. Parts kits include sockets for all ICs, components, and circuit board. Documentation is included with all products. All items are in stock, and will be shipped the day order is received via first class mail. Prices are in US dollars. No open accounts. To eliminate tariff in Canada boxes are marked "Computer Parts." Dealer inquiries invited. 24 Hour Order Line: (408) 226-4064

* Circuits designed by John Bell

DIODES/ZENERS

1N914	100v	10mA	.05
1N4005	600v	1A	.08
1N4007	1000v	1A	.15
1N4148	75v	10mA	.05
1N4733	5.1v	1 W Zener	.25
1N753A	6.2v	500 mW Zener	.25
1N758A	10v	"	.25
1N759A	12v	"	.25
1N5243	13v	"	.25
1N5244B	14v	"	.25
1N5245B	15v	"	.25

SOCKETS/BRIDGES

8-pin	pcb	.20	ww	.35
14-pin	pcb	.20	ww	.40
16-pin	pcb	.20	ww	.40
18-pin	pcb	.25	ww	.75
22-pin	pcb	.35	ww	.95
24-pin	pcb	.35	ww	.95
28-pin	pcb	.45	ww	1.25
40-pin	pcb	.50	ww	1.25

Molex pins .01 To-3 Sockets .25
2 Amp Bridge 100-prv .95
25 Amp Bridge 200-prv 1.95

TRANSISTORS, LEDS, etc.

2N2222	NPN (2N2222 Plastic .10)	.15
2N2907	PNP	.15
2N3906	PNP (Plastic - Unmarked)	.10
2N3904	NPN (Plastic - Unmarked)	.10
2N3054	NPN	.35
2N3055	NPN 15A 60v	.50
T1P125	PNP Darlington	.35
LED Green, Red, Clear, Yellow		.15
D.L.747	7 seg 5/8" High com-anode	1.95
MAN72	7 seg com-anode (Red)	1.25
MAN3610	7 seg com-anode (Orange)	1.25
MAN82A	7 seg com-anode (Yellow)	1.25
MAN74A	7 seg com-cathode (Red)	1.50
FND359	7 seg com-cathode (Red)	1.25

C MOS

4000	.15
4001	.15
4002	.20
4004	3.95
4006	.95
4007	.20
4008	.75
4009	.35
4010	.35
4011	.20
4012	.20
4013	.40
4014	.75
4015	.75
4016	.35
4017	.75
4018	.75
4019	.35
4020	.85
4021	.75
4022	.75
4023	.20
4024	.75
4025	.20
4026	1.95
4027	.35
4028	.75
4030	.35
4033	1.50
4034	2.45
4035	.75
4040	.75
4041	.69
4042	.65
4043	.50
4044	.65
4046	1.25
4049	.45
4050	.45
4066	.55
4069/74C04	.25
4071	.25
4081	.30
4082	.30
MC 14409	14.50
MC 14419	4.85
4511	.95
74C151	1.90

– T T L –

7400	.10
7401	.15
7402	.15
7403	.15
7404	.10
7405	.25
7406	.25
7407	.55
7408	.15
7409	.15
7410	.15
7411	.25
7412	.25
7413	.25
7414	.75
7416	.25
7417	.40
7420	.15
7426	.25
7427	.25
7430	.15
7432	.20
7437	.20
7438	.20
7440	.20
7441	1.15
7442	.45
7443	.45
7444	.45
7445	.65
7446	.70
7447	.70
7448	.50
7450	.25
7451	.25
7453	.20
7454	.25
7460	.40
7470	.45
7472	.40

7473	.25
7474	.30
7475	.35
7476	.40
7480	.55
7481	.75
7483	.75
7485	.55
7486	.25
7489	1.05
7490	.45
7491	.70
7492	.45
7493	.35
7494	.75
7495	.60
7496	.80
74100	1.15
74107	.25
74121	.35
74122	.55
74123	.35
74125	.45
74126	.35
74132	.75
74141	.90
74150	.85
74151	.65
74153	.75
74154	.95
74156	.70
74157	.65
74161	.55
74163	.85
74164	.60
74165	1.10
74166	1.25
74175	.80

74176	.85
74180	.55
74181	2.25
74182	.75
74190	1.25
74191	.95
74192	.75
74193	.85
74194	.95
74195	.95
74196	.95
74197	.95
74198	1.45
74221	1.00
74367	.75
75108A	.35
75491	.50
75492	.50
74H00	.15
74H01	.20
74H04	.20
74H05	.20
74H08	.35
74H10	.35
74H11	.25
74H15	.45
74H20	.25
74H21	.25
74H22	.40
74H30	.20
74H40	.25
74H50	.25
74H51	.25
74H52	.15
74H53J	.25
74H55	.20

74H72	.35
74H101	.75
74H103	.55
74H106	.95
74L00	.25
74L02	.20
74L03	.25
74L04	.30
74L10	.20
74L20	.35
74L30	.45
74L47	1.95
74L51	.45
74L55	.65
74L72	.45
74L73	.40
74L74	.45
74L75	.55
74L93	.55
74L123	.85
74S00	.35
74S02	.35
74S03	.25
74S04	.25
74S05	.35
74S08	.35
74S10	.35
74S11	.35
74S20	.25
74S40	.20
74S50	.20
74S51	.25
74S64	.15
74S74	.35
74S112	.60
74S114	.65

74S133	.40
74S140	.55
74S151	.30
74S153	.35
74S157	.75
74S158	.30
74S194	1.05
74S257 (8123)	1.05
74LS00	.20
74LS01	.20
74LS02	.20
74LS04	.20
74LS05	.25
74LS08	.25
74LS09	.25
74LS10	.25
74LS11	.25
74LS20	.20
74LS21	.25
74LS22	.25
74LS32	.25
74LS37	.25
74LS38	.35
74LS40	.30
74LS42	.65
74LS51	.35
74LS74	.35
74LS86	.35
74LS90	.55
74LS93	.55
74LS107	.40
74LS123	1.00
74LS151	.75
74LS153	.75
74LS157	.75
74LS164	1.00
74LS193	.95
74LS367	.75
74LS368	.65

9000 SERIES

9301	.85	95H03	1.10
9309	.35	9601	.20
9322	.65	9602	.45

MICRO'S, RAMS, CPU'S, E-PROMS

74S188	3.00	8214	8.95
1702A	4.50	8224	3.25
MM5314	3.00	8228	6.00
MM5316	3.50	8251	8.50
2102-1	1.45	8255	10.50
2102L-1	1.75	8T13	1.50
2114	9.50	8T23	1.50
TR1602B	3.95	8T24	2.00
TMS 4044-	9.95	8T97	1.00
		2107B-4	4.95
8080	8.95	2708	9.50
8212	2.95	Z80 PIO	8.50

LINEARS, REGULATORS, etc.

MCT2	.95
8038	3.95
LM201	.75
LM301	.45
LM308 (Mini)	.95
LM309H	.65
LM309K (340K-5)	.85
LM310	.85
LM311D (Mini)	.75
LM318 (Mini)	1.75
LM320K5(7905)	1.65
LM320K12	1.65

LM320T5	1.65
LM320T12	1.65
LM320T15	1.65
LM324N	1.25
LM339	.75
7805 (340T5)	.95
LM340T12	.95
LM340T15	.95
LM340T18	.95
LM340T24	.95
LM340K12	1.25

LM340K15	1.25
LM340K18	1.25
LM340K24	1.25
78L05	.75
78L12	.75
78L15	.75
78M05	.75
LM373	2.95
LM380 (8-14 PIN)	.95
LM709 (8, 14 PIN)	.25
LM711	.45

LM723	.40
LM725N	2.50
LM739	1.50
LM741 (8-14)	.25
LM747	1.10
LM1307	1.25
LM1458	.65
LM3900	.50
LM75451	.65
NE555	.35
NE556	.85
NE565	.95
NE566	1.25
NE567	.95

INTEGRATED CIRCUITS UNLIMITED

19

7889 Clairemont Mesa Boulevard, San Diego, California 92111
(714) 278-4394 *(Calif. Res.)*

All orders shipped prepaid — No minimum
Open accounts invited — COD orders accepted

Discounts available at OEM Quantities — California Residents add 6% Sales Tax
All IC's Prime/Guaranteed. All orders shipped same day received.

24 Hour Toll Free Phone 1-800-854-2211 — **American Express / BankAmericard / Visa / MasterCharge**

SPECIAL DISCOUNTS

Total Order	Deduct
\$35 - \$99	10%
\$100 - \$300	15%
\$301 - \$1000	20%

ELECTRONICS EMPORIUM INTERNATIONAL 19

7895 Clairemont Mesa Blvd. • San Diego, California 92111

24 Hour Toll Free Phone 1-800-854-2211 (714) 278-4394 *(Calif. Res.)*

All orders shipped prepaid • COD orders accepted

All orders shipped same day received

California residents at 6% sales tax

Master Charge / BankAmericard / AE

ASTRON POWER SUPPLIES

HEAVY DUTY • HIGH QUALITY • RUGGED • RELIABLE •

PECIAL FEATURES

- SOLID STATE ELECTRONICALLY REGULATED
- FOLD-BACK CURRENT LIMITING Protects Power Supply from excessive current & continuous shorted output.
- CROWBAR OVER VOLTAGE PROTECTION on Models RS-6A, RS-12A, RS-20A, & RS-35A.
- MAINTAIN REGULATION & LOW RIPPLE at low line input Voltage.
- HEAVY DUTY HEAT SINK
- CHASSIS MOUNT FUSE
- THREE CONDUCTOR POWER CORD
- ONE YEAR WARRANTY
- MADE IN U.S.A.

ERFORMANCE SPECIFICATIONS

- INPUT VOLTAGE: 105 - 125 VAC
- OUTPUT VOLTAGE: 13.8 VDC ± 0.05 volts
- RIPPLE: Less than 5mv peak to peak (full load & low line)
- REGULATION: ± .05 volts no load to full load & low line to high line

$129.95

ASTRON 35 AMP POWER SUPPLY
Model RS-35A

25 Amps continuous
35 Amps ICS
5″ (H) x 11″ (W) x 11″ (D)

ASTRON 20 AMP POWER SUPPLY
Model RS-20A

16 Amps continuous
20 Amps ICS
5″ (H) x 9″ (W) x 9″ (D)

$89.95

ASTRON 12 AMP REGULATED POWER SUPPLY
Model RS-12A

9 Amps continuous
12 Amps ICS
4″ (H) x 8″ (W) x 9″ (D)

$69.95

ASTRON 7 AMP REGULATED POWER SUPPLY
Model RS-7A

5 Amps continuous
7 Amps ICS

$49.95

ASTRON 4 AMP REGULATED POWER SUPPLY
Model RS-4A

3 Amps continuous
4 Amps ICS

$39.95

$88 CONTROL

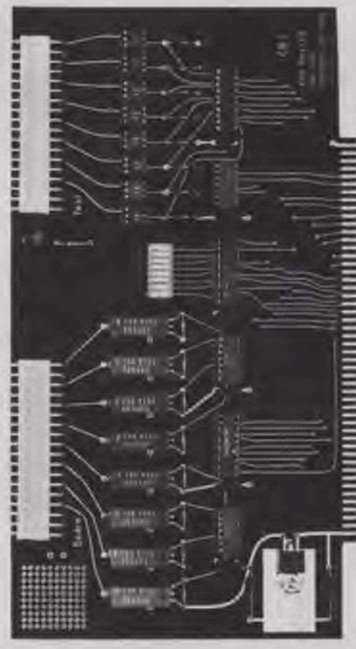
People have used our original model CB-0 Controller Board for controlling audio systems, model railroads, time lapse photography, and dozens of other applications requiring intelligent, computer-controlled switching.

Our improved model CB-1 has all the features of its predecessor: 8 relays that respond to an 8 bit word for control purposes, 8 opto-isolators that accept input data for handshaking or further control purposes, full S-100 buss compatibility, address selection switch, quality components and board, and so on.

But . . . the limited use (and expensive!) flat cable connector has been replaced with new connectors; these allow you to use pairs of wires of mixed gauges and lengths as needed. Also, a self-test feature is built on the board itself, and a new output design allows replacement of the relays with opto-isolators.

The best part is that you don't pay a penny more for these new features. In fact, compared to the CB-0 price, you pay 2,900 pennies less.

Available at computer stores nation-wide, or by direct mail (mail orders shipped postpaid in USA; Californians add sales tax).



The Mullen CB-1 Controller Board--- now at a new low price.

M32 **MULLEN COMPUTER BOARDS**
BOX 6214, HAYWARD, CA 94545

THE OE 1000 IS NOT CHEAP, ONLY INEXPENSIVE

The OE is supplied with only prime components from major manufactures. It uses a solder masked, component marked, pc board that is double sided with plated through holes, and a gold plated edge connector. All IC's have sockets and the unit contains its' own power supply. The video output is composite, ready for input into a modified TV or video monitor. The OE 1000 will display 96 ASCII characters plus 32 special characters (greek, math and other symbols) on 16 lines of 64 characters. The keyboard used will output both upper and lower case or just upper case (TTY mode). The terminal accepts data in RS 232 or 20 ma current loop form and has selectable baud rates of 110 or 300 baud. The OE 1000 is housed in a large, high impact, plastic case, leaving enough room for expansion of your microcomputer. All this for only $275.00 in kit and $350.00 assembled. (add $5.00 for insurance and shipping cost) YES we are inexpensive, but not cheap.

MC, BAC/VISA, COD ACCEPTED 609 448 9165 NJ RESIDENTS ADD 5% SALES TAX

OTTO ELECTRONICS, PO BOX 3066, PRINCETON, NJ 08540 O9

ADVANCED COMPUTER PRODUCTS A38

COMPUTER SPECIALS

WE'VE GOT A TRUCKLOAD COMING!
. . . and it's COLOR
"The Compucolor II"

. . . a personal colorgraphics system for the modern computer man . . .

- ★ Color Graphics 13" Color CRT
- ★ Proven 8080A CPU System
- ★ 16K Extended Disk Basic
- ★ Up to 117* Key Keyboard
- ★ Up to 32K* RAM
- ★ Minidisk Drive 51.2K Bytes/Side

*options

Order your system now and we will include at no cost (2) Programmed Software Diskettes ($39.90 value)

Model 3 with 8K RAM 71 Key Keyboard.............. **$1395**
Model 4 with 16K RAM & RS232C port plus 71 Key Keyboard...$1795.00
Model 5 with 32K RAM, RS232C and 71 Key Keyboard $1995.00
Options: for RS232 on Model 3....................... Add $100.00
101 Key Keyboard .... Add $150.00
117 Key Keyboard .... Add $225.00
Programmed Diskettes.............................. $19.95 each
Formatted Diskettes...................................... 2/$19.95

EXIDY SORCERER™

$895.00

User programmable or use program cartridges. Combines the desirable features of the PET, APPLE and TRS-80 into a complete expandable computer system. If you are a comparison shopper send for comparison chart.

★ INCLUDES:
Keyboard & enclosure totally assembled
90 day Warranty
19.5K BASIC (MICROSOFT)
Video & Cassette Cable
Complete Documentation
★★ S100 Expansion Module ..... Add $299.00
★★ Cassette recorder .... Add $44.95
★★ Sanyo 9" Monitor .... Add $159.95

KIM-1

Now only
$229.95

Regular price $245.00

The KIM-1 6502 Standard Computer System now available off the shelf. Write for complete details and software package.

★★ Power Supply .... Add $95.00
★★ Cassette Recorder .... Add $44.95
★★ Sanyo 9" Monitor .... Add $159.95

VIM-1

"The Ultimate in Single Board Low-Cost Computers" In Stock

$269.00

- ★ KIM-1 Compatible
- ★ 4K ROM Monitor
- ★ 1K Bytes 2114 RAM
- ★ 65K Memory Expansion
- ★ User EPROM 2716

★★ Power Supply .... Add $95.00
★★ Cassette Recorder .... Add $44.95
★★ Sanyo 9" Monitor .... Add $159.95

School & group discounts available.

AIM-1

NEW FROM ROCKWELL INTERNATIONAL

Singleboard Computer
- ★ On Board 20 column alphanumeric printer
- ★ Alphanumeric 20 character display
- ★ Terminal style Keyboard 54 Keys
- ★ 6502 based CPU

w/1K RAM $375.00
w/4K RAM $450.00
Assembler ROM ... Add $85.00
BASIC Interpreter in ROM .... Add $100.00

RCA COSMAC VIP

NEW LOW PRICE **$249.00**
Assembled
Regular price $299.95
w/Sanyo 9" Monitor .... Add $159.95

"Now you can afford to experiment using RCA's fine 1802 CMOS CPU."

BALLY ARCADE

Z-80 based expandable to 48K
Color Display BASIC available in ROM Cartridge
8K ROM
AK RAM

$299.95

NORTH STAR HORIZON

Now in stock North Star Z-80 based high-performance computer.

- ★ Z-80 Processor
- ★ Motherboard
- ★ 2 Serial +1 Parallel Port
- ★ 16K RAM

Horizon I .... $1439.00 Kit
Horizon II.... $1799.00 Kit

PRINTERS

OKIDATA Model 110 w/tractor w/RS232 .... $1675.00
OKIDATA Model 22 w/tractor w/RS232 .... $2705.00
DECWRITER II w/RS232 10-30 cps $1475.00
DIABLO 1620-3 w/tractor feed, w/Keyboard........ $3255.00
DIABLO 1610-3 w/tractor feed $2995.00
IPSI 1622-3 w/tractor feed (diablo compatible)........ $2995.00
IPSI 1612-3 w/tractor feed (diablo compatible)........ $2820.00
Centronics 761 (KSR) $1595.00
Centronics 761RO $1495.00
Centronics 779 w/tractor feed ... $1195.00
Teletype Model 43 w/RS232 ... $1199.00
TI Model 745 w/upper and lower case $1975.00

FLOPPY DISK'S

Shugart SA400 Minifloppy Disk Drive $325.00
Shugart SA450 Dual Density Minifloppy $375.00
Shugart SA800/801 Diskette Storage Drive $495.00
Shugart SA850/851 Double-Sided Drive........ $625.00
MFE Mayflower Model 751 Double Density........ $730.00
Persci Model 277 Dual Diskette Storage Drive $1230.00
Pertec FD200 Minifloppy Disk Drive $299.00
Calcomp Model 143M Dual Density Drive........ $625.00
North Star MDS Minifloppy w/S-100 .... $599.00

TERMINALS

SOROC IQ 120........ $895.00
LEAR ADM3 Assembled $829.95
LEAR ADM3 Kit $739.95
Sanyo 9" Monitor........ $159.95
Sanyo 15" Monitor........ $209.95
Motorola 12" Monitor w/o chassis $219.95
Hazeltine 1500........ $1095.00
Hazeltine 1510........ $1295.00

KEYBOARDS

GEO Risk Model 756 ASCII 56 Key Assembled $67.95
Metal case for Model 756........ $27.00
Clare Pender 62 Key ASCII w/26 Pin and 34 Pin Connector (new surplus supply limited)...... $54.95
63 Key Unencoded Keyboard $32.95
10 Key Hexpad Unencoded $11.95

ADVANCED COMPUTER PRODUCTS
P. O. Box 17329 • Irvine, California 92713
1310 B E. Edinger • Santa Ana, California 92705
TWX/TELEX 910 595 1565 • TEL (714) 558-8813

All prices FOB Santa Ana. Please add shipping/handling. For credit card purchase contact for quotation. Complete FREE 1978 Catalogue – send 25¢ for postage.

16K E-PROM CARD

New!

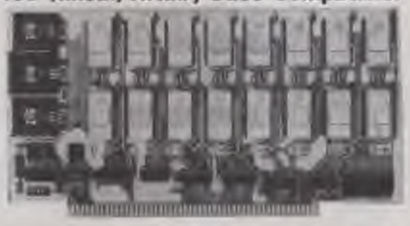
IMAGINE HAVING 16K OF SOFTWARE ON LINE AT ALL TIME!

S-100 (Imsai/Altair) Buss Compatible!

KIT FEATURES:

1. Double sided PC board with solder mask and silk screen and gold plated contact fingers.
2. Selectable wait states.
3. All address lines & data lines buffered!
4. All sockets included.
5. On card regulators.

KIT INCLUDES ALL PARTS AND SOCKETS (except 2708's). Add $25. for assembled and tested.



PRICE CUT!

$57.50 kit

WAS $69.95

SPECIAL OFFER:

Our 2708's (450NS) are $12.95 when purchased with above kit.

8K LOW POWER RAM KIT-$149.00

Fully Static!

ADD $20 FOR 250NS

S-100 (Imsai/Altair) Buss Compatible!

KIT FEATURES:

1. Doubled sided PC Board with solder mask and silk screen layout. Gold plated contact fingers.
2. All sockets included.
3. Fully buffered on all address and data lines.
4. Phantom is jumper selectable to pin 67.
5. FOUR 7805 regulators are provided on card.

(450NS)

USES 21L02 RAM'S!

2 KITS FOR $279

Fully Assembled & Burned In
$179.00

Blank PC Board w/ Documentation
$29.95

Low Profile Socket Set 13.50

Support IC's (TTL & Regulators)
$9.75

Bypass CAP's (Disc & Tantalums)
$4.50

MOTOROLA QUAD OP - AMP
MC 3401. PIN FOR PIN SUB. FOR POPULAR LM 3900.
3 FOR $1

ALARM CLOCK CHIP
N.S. MM5375AA. Six Digits. With full Data. **New!**
$1.95 each

FULL WAVE BRIDGE
4 AMP. 200 PIV.
69¢ 10 FOR $5.75

NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE SUPPLIERS OF CPM SOFTWARE.

MOTOROLA 7805R VOLTAGE REGULATOR
Same as standard 7805 except 750 MA output. TO–220. 5VDC output.
44c each or 10 for $3.95

450 NS! 2708 EPROMS
Now full speed! Prime new units from a major U.S. Mfg. 450 N.S. Access time. 1K x 8. Equiv. to 4-1702 A's in one package.
$15.75 ea. 4 FOR $50.00

INTEL 2102 RAM SALE!

BRAND NEW 2102A-4. FACTORY PRIME!

WE MADE ANOTHER SUPER SURPLUS BUY!

PERFECT FOR USE WITH OUR BLANK 8K PC BOARD SHOWN ABOVE.

THESE PARTS HAVE BEEN **SCREENED** TO MEET THE LOW POWER SPEC. AND ARE GUARANTEED BY US TO BE 40 MA. MAXIMUM ICC.
(28 MA. TYPICAL)

LOW POWER

450 N.S.

8 FOR $7.95

32 FOR $28

"A" VERSION FOR BATTERY BACKUP!

4K STATIC RAM'S
2114. The new industry standard. Arranged as 1K x4. Equivalent to 4-21 L02's in 1 package! 18 pin DIP. 2 chips give 1Kx8.
2/$24. 8/$85.

OPCOA LED READOUT
SLA-1. Common Anode. .33 inch character size. The original high efficiency LED display. **75c ea.**
4 FOR $2.50

Z-80 PROGRAMMING MANUAL
By Mostek. The major Z-80 second source. The most detailed explanation ever on the working of the Z-80 CPU CHIPS. At least one full page on each of the 158 Z-80 instructions. A MUST reference manual for any user of the Z-80. 300 pages. Just off the press. **$12.95**

NATIONAL SEMICONDUCTOR

JUMBO CLOCK MODULE

MA1008A BRAND NEW!

$6.95

2 FOR $13

(AC XFMR $1.95)

FEATURES:
- FOUR JUMBO ½ INCH LED DISPLAYS
- 12 HR REAL TIME FORMAT
- 24 HR ALARM SIGNAL OUTPUT
- 50 OR 60 HZ OPERATION
- LED BRIGHTNESS CONTROL
- POWER FAILURE INDICATOR
- SLEEP & SNOOZE TIMERS
- DIRECT LED DRIVE (LOW RFI)
- COMES WITH FULL DATA

ASSEMBLED! NOT A KIT!

COMPARE AT UP TO TWICE OUR PRICE!

ZULU VERSION!
We have a limited number of the 24 HR Real time version of this module in stock.
#MA1008D — $9.95

PERFECT FOR USE WITH A TIMEBASE.

MANUFACTURER'S CLOSEOUT!

SALE!
1N4148 DIODES. SILICON. Same as 1N914. New, factory prime, Full Leads.
100 FOR $2
1000 FOR $17.50

New! REAL TIME Computer Clock Chip
N.S. MM5313. Features BOTH 7 segment and BCD outputs. 28 Pin DIP. **$4.95 with Data**

MICRO-MINI TOGGLE SWITCH

SPDT. By RAYTHEON. MADE IN USA! WITH HDWR.

99¢ EACH

6 FOR $5

Digital Research Corporation
(OF TEXAS)

P. O. BOX 401247 • GARLAND, TEXAS 75040 • (214) 271-2461

TERMS: Add 30¢ postage, we pay balance. Orders under $15 add 75¢ handling. No C.O.D. We accept Visa, Mastercharge, and American Express cards. Tex. Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. 90 Day Money Back Guarantee on all items.

ADVANCED COMPUTER PRODUCTS

A38

THE FIRST TO OFFER PRIME PRODUCTS TO THE HOBBYIST AT FAIR PRICES NOW LOWERS PRICES EVEN FURTHER!

1. Proven Quality Factory tested products only, no re-tests or fallouts. Guaranteed money back. We stand behind our products.

2. Same Day Shipment All prepaid orders with cashiers check, money order or charge card will be shipped same day as received.

S100 PRODUCTS

LOGOS I 8K STATIC RAM
- ★ Low Power
- ★ Selectable Memory Protect
- ★ Totally Buffered
- ★ Battery Back-up
- ★ Address on 1K boundary
- ★ Requires no front panel
- ★ No wait states

ASSEMBLED & TESTED
250ns. 199.95
450ns. 179.95

KIT
250ns. 149.95
450ns. 125.95

WE OFFER GROUP DISCOUNTS

Bare PC Board w/Data **$21.95**
Now over 1 year successful field experience
"Special Offer" Buy (4) 8K 450ns. Kits $117.00 ea.

IMS 16K STATIC RAM
- ★ Memory mapping
- ★ Phantom
- ★ Address 16K Bound
- ★ Use with or w/o front panel
- ★ Power 2.1 amps typ.
- ★ Uses 2114

ASSEMBLED & TESTED ONLY
450ns. $439.95 250ns. $495.95

S-100 32K STATIC RAM
- ★ Address 32K Boundary
- ★ Power 450ns. 2.8 amps typ.
- ★ No wait states on 2MHz
- ★ Fully Buffered
- ★ Phantom can be added
- ★ 2114/TMS4045 or 9135

ASSEMBLED & TESTED	KIT
250ns. $849.00	250ns. $790.00
450ns. $799.00	450ns. $740.00

Bare PC Board w/Data $59.95
SPECIAL OFFER:
Kit without Memory only $99.95. Add 2114 Memory for $160.00 per 8K

PARATRONICS LOGIC ANALYZER KIT
MODEL 100A $219.95
(analyzes any type of digital system)
Trigger Expander Model 10 $229.00
Baseplate 9.95
Model 10 Manual 4.95
Model 150 Bus Grabber Kit 369.00

DC HAYES DATA COMMUNICATIONS ADAPTER
★ Telephone/TWX ★ S-100 compatible
★ Bell 103 freq. ★ Originate & answer mode
Assembled & Tested $279.95
Bare PC Board w/data $ 49.95

Z-80/Z-80A CPU BOARD
- ★ On board 2708
- ★ Power on jump
- ★ 2708 included (450ns.)
- ★ completely socketed

Assembled and tested $185.00
Kit $129.95
Bare PC Board $ 34.95
★ For 4MHz Speed Add $15.00

TDL COMPATIBLE Z-80 CPU
★ Compatible to TDL software & hardware ★ Can be upgraded for 4MHz operation
Assembled and tested $199.95
Kit $139.95
Bare PC Board $ 34.95

BYTE USER 8K EPROM BOARD
★ Power on Jump ★ Reset Jump
Assembled & Tested 94.95
Kit 64.95
Bare Pc Board 21.95
Special Offer: Buy 4 kits only $59.95 each
NOTE: 2708-6 only $5.95

TRS 80 UPGRADE KIT
16K Memory with Jumpers and Instructions $159.95
(Specify Level I or Level II)
APPLE II 16K UPGRADE KIT $149.95

TARBELL FLOPPY INTERFACE
★ S100 Compatible ★ Uses CPM
★ Jumper Selectable ★ Persci,Shugart,etc
Assembled and tested $269.95
Kit $179.95
Bare PC Board $ 39.95
NOTE: For CPM Add $70.00
Documentation Add $20.00

PET TO S-100 ADAPTER
Allows Pet to be interfaced to popular S-100 Bus.
Kit $189.95
Assembled $269.95
For Low Cost 8 Slot Bus to expand your Pet only $149.95 Kit

MICRODESIGN MR-16 2716 EPROM BOARD (MR-8 Also Available at same price)
★ Individual Prom Address
★ Uses Low cost 16K TI EPROMS
★ Optional 1K RAM ★ Phantom-control
Assembled and Tested 174.95
Kit 99.50

DATABOOKS & MANUALS

NSC TTL Data	$ 3.95	AMI MOS/LSI Data	3.95	Mot. Vol. 4 Mect Data	3.95
NSC Linear Data	4.95	GI MOS/LSI Data	4.95	Mot. Vol. 5 CMOS Data	2.95
NSC Linear APP Note I	3.95	Osborne Intro to Micro Vol. 0	7.50	Mot. Vol. 6 Linear Data	3.95
NSC Linear APP Note II	3.95	Osborne Intro to Micro Vol. I	7.50	Mot. Vol. 9 Schottky TTL	2.95
NSC CMOS Data	2.95	Osborne Intro to Micro Vol. II	7.50	Mot. MPU Applications	25.00
NSC Audio Data	3.95	Osborne 8080 Programming	7.50	Mot. MPU Prog. Ref. Manual	3.00
NSC Volt. Reg. Data	2.95	Osborne 6800 Programming	7.50	Mot. Power Data	2.95
NSC Memory Data	3.95	Osborne Z80 Programming	7.50	Mot. Rectifier Data	2.95
NSC MOS/LSI Data	3.95	TI Power Semi's Data	7.50	Mot. Switching Tran.	2.95
NSC Power Transistors	2.95	TI TTL Data	6.95	Mot Zenors	2.95
Intel Databook	3.95	TI Transistors & Diodes	6.50	Basic Software SRI Vol. I or II	24.95
Intel MCS85 Manual	7.50	TI Memory Data	3.95	Basic Software SRI Vol. III	39.95
Intel MCS80 Manual	7.95	TI Optoelectronics	3.95	Basic Software SRI Vol. IV or V	9.95
Intel MCS40 Manual	4.95	TI Linear Data	3.95	Basic Software SRI Vol. VI	49.95
AMD 8080 Handbook	3.95	TI Bipolar Memory	3.95	Basic Software SRI Vol. VII	39.95
AMD Linear Data	3.95	TI Interface Data	4.95	1978 IC Master	47.50
AMD Schottky Data	4.95	Motorola Semi Data 1, 2, 3	8.95		

6800 DESIGNER BOARDS MODULES PROTO BOARDS

★ Motorola Compatible Modules*

MEK 6800 D2 Kit	$235.00
9600 6800 MPU Module	495.00
9601 16 slot Mother Bd.	175.00
9602 16 slot Card Cage	75.00
9603 8 slot Mother Bd.	99.00
9604 System Power Supply	250.00
9610 Prototype Board	36.00
9615 4KEprom Module	250.00
9620 16 port parallel I/O	375.00
9626 8K Static RAM	295.00
9626K 8K Static RAM Kit	225.00
9630 Extender Card	60.00
9640 Multiple Tuner Prog.	395.00
9650 8 port Duplex Asyn. I/O	395.00
Mot 43/86 Connectors w/w or s/t	5.95
AMI EVK 99* 6800 sub Kit	99.00
AMI EVK 200 Kit	249.95
AMI EVK 300 Assembled	275.00
EVK Kluge Board	95.00
EVK 16K Byte Ram Board	75.00
EVK 8 Slot Motherboard	35.00
EVK Extender Board	45.00
EVK Solid Frame Chassis	129.00
EVK Connectors	6.50
AMI 6800 Proto Rom	30.00
AMI 6800 Micro Assembler Rom	30.00
6800 Tiny Basic Paper Tape	20.00
6800 Tiny Basic Eprom	125.00

ZILOG COMPATIBLE BOARDS

Z-80 CPU	395.00
ZDC Disc Controller	395.00
MEM 16/65K Memory	595.00

SUPPORT DEVICES

AM9511 Arith. Processor	$195.00
AM9517 DMA Controller	71.95
AM9519 Universal Interrupt	24.95
3881 (Z80 PIO)	11.25
3881-4 (4MHz)	14.95
3882 (Z-80 CTC)	11.25
3882-4 (4MHz)	14.95
8205/74S138 Decoder	2.95
8212 8 bit I/O	3.25
8214 Priority Int	7.95
8216 Bus Driver	3.25
8224 Clock Gen	3.95
8224-4 (4MHz)	9.75
8226 Bus Driver	3.95
8T26 Bus Driver	2.39
8228 Sys. Control	6.95
8238 Sys. Cont	7.45
8251 Prog. I/O	11.50
8253 Int. Timer	23.95
8255 Prog. I/O	11.50
8257 Prog. DMA	27.95
8259 Prog. Int	27.95
8275 CRT Controller	125.00
8279 Prog. Keyboard	19.95
6810-1 128 x 8 RAM	4.75
6820 PIA	7.95
6828 Priority Int.	11.95
6834-1 512 x 8 Eprom	12.95
6850 ACIA	7.95
6852 Serial Adapter	9.95
6854/HD46505 CRT Cont.	69.95
6860 Modem	14.50
6862 Modulator	17.50
6871A 1.0MHz OSC	25.95
2350 USRT	9.95
8880 Bus Driver	2.39
1821 SCD 1K RAM	25.00
1822 SCD 256 x 4 RAM	19.95
1824 CD 32 x 8 RAM	9.85
1852 CD 8 bit I/O	10.95
1856 CD I/O	8.95
1857 CD I/O	8.95
6520 PIA	8.95
6522 Mult	9.95
6530-002	15.50
6530-003	15.50
6530-004	15.50
6530-005	15.50
3851 F8 Prog. Store	14.95
3853 F8 Memory I/O	14.95

PROMS

2708	12.95
2708-6	5.95
1702A	4.95
1702A-6	3.75
2716-5V	44.95
2716-5V, 12V	24.95
2758 5V	25.95
5203AQ	11.95
5204AQ	8.95
6834-1	12.95
IM 5610	2.95
82S123	2.95
82S126	4.50
82S129	4.95

RAMS DYN.

4116 16K x 1	18.95
4115 8K x 1	11.95
4050 4K x 1	4.45
4060 4K x 1	4.45
4096 4K x 1	4.25
2104 4K x 1	4.50
2107B-4 4K x 1	4.25
5261	1.95
5262	1.95
5270	4.90
5280	4.95
1103	1.95
4008L	4.95
TMS4070 16K x 1	19.95
4027-3	4.50
MCM6605 4K x 1	4.95

FLOPPY DISK I/O

1771 I/O 8" only	22.95
1771-01 8" & Minifloppy	27.95
µPD372 Nec Floppy	49.95
1781 Dual Floppy	89.95

CHARACTER GENERATORS

RO32513-001 (5V) Upper	9.50
RO32513-002 (5V) Lower	10.95
RO32513-ADM3 (5V) Lower	14.95
MCM6571	10.75
MCM6571A	10.75
MCM6574	14.50
MCM6575	14.50

KEYBOARD ENCODERS

AY5-2376	13.75
AY5-3600	13.75

UARTS/USRTS

TR1602B (5V, 12V)	3.95
AY51013 (5V, 12V)	5.50
AY51014A/1612 (5-14V)	6.95
AY51015A/1863 (5V)	6.95
TMS 6011 (5V, 12V)	6.50
IM6402	9.95
IM6403	9.95
2350 USRT	9.95
1671B Astro	24.95

BAUD RATE GENERATORS

MC14411	11.25
4702	14.95
WD1941	9.95

TV CHIPS

MM5320 TV Synch; MM5369 Prescaler; LM1889 Modulator; CA900 Sine Function; MM57100 Color TV Game; MM57104 Clock; AY38500-1 TV Game; AY38600 Color TV Game; AY38615-1 Color Converter; AY38700-1 Tank Chip; RF Modulator with Audio

A/D CONVERTERS

8700 8 bit Binary; 8701 10 bit Binary; 8702 12 bit Binary; 8703 12 bit TS; 9400 Volt to Freq. Conv.; 8750 3-1/2 Digit BCD; 8038 Multifunction; 1408L6 6 bit; 1408L8 8 bit; DAC01 A/D

TV INTERFACES

Pixie-Verter; TV-1 Video Interface; Pickles & Trout TV Mod. Kit 20.00; Multiverter 35.00; M&R Modulator 35.00

MICROPROCESSORS

Z-80	$21.95
Z-80A	29.95
F-8 (3850)	19.95
2650	24.95
CD1802	19.95
8080A	12.95
SALE 8085	**18.95**
8008-1	14.95
2901	21.95
2901A	29.95
TMS 9900/L	69.95
CP1600	89.95
6502	19.95
IM6100	29.95
6800	**19.95**
6802P	27.95

SOCKETS

8 Pin ww	.32	14 Pin S/T	.20
14 Pin ww	.37	16 Pin S/T	.22
16 Pin ww	.38	18 Pin S/T	.31
18 Pin ww	.50	22 Pin S/T	.35
20 Pin ww	.80	24 Pin S/T	.41
22 Pin ww	.83	28 Pin S/T	.49
8 Pin S/T	.17	40 Pin S/T	.63

CONNECTORS

8 Pin Single S/T	1.49
15/30 Dual S/E	1.95
18/36 Dual S/E	2.35
22/44 Dual S/E	2.95
40/80 W/W	4.95
43/86 Dual W/W/S/T	6.50
50/100 IMSAI W/W	4.75
50/100 IMSAI S/T	4.75
50/100 Altair W/W/S/T	5.95
IMSAI Card Guides	8/1.00

25 Pin D Subminiature	
DB25P	3.25
DB25S	3.75
Hood	1.25
Set w/Hood	6.50

NEW CTS DIPSWITCHES

CTS206-4	$1.75	CTS206-8	$1.95
CTS206-5	$1.75	CTS209-9	$1.95
CTS206-6	$1.75	CTS206-10	$1.95
CTS207-7	$1.75		

8080 USER'S IT'S HERE LIVERMORE BASIC
ON 8SK ROM
THAT'S RIGHT ON ROM!!
OUR PRICE only $95.00

PCS PROGRAMMABLE CHARACTER GENERATOR
The hottest version of STARWARS available
S-100 Compatible 2MHz kit with object code on tarbell or cuts $169.95

NAKED PC BOARD SALE S-100

Z-80 CPU (ITHACA)	34.95
Z-80 CPU (TDL comp.)	34.95
8080A CPU	34.95
8K Static Logos	21.95
8K Eprom Bytesaver	21.95
32 Static S-100	59.95
Tarbell Floppy I/O	39.95
Tarbell Cassette	29.95
Vector 8800 Proto	19.95
ACP Proto Bd.	27.95
ACP Extender w/conn	15.95
Realtime Clock	34.95
WMC 13 Slot Mother Bd.	33.00
S-100 8 Slot Mother Bd	34.95
Vector 8803 11 Slot M. Bd.	29.95
SSM Video I/O	27.95
SSM I/O2 Parallel I/O	27.95

UV "Eprom" Eraser
Model UVS-11E $59.50
Holds 4 Eprom's at a time
Backed by 45 years UV experience
Model S-52T $219.95
Professional Industrial Model

ATTENTION KIM USERS

ATTENTION PET USERS

ATTENTION TRS-80 USERS

ATTENTION APPLE II USERS

All Shipments FCM or UPS. Orders under $100.00 add 5% handling and postage. Orders over $100.00 add 2.5% handling & postage. Mastercharge/BankAmericard/COD accepted w/25% deposit. California Residents add 6% tax. Foreign Orders add 6% handling. All parts guaranteed. Add 25 cents for Data.

Actual pricing may vary from Mail Order Pricing

STATIC RAM HEADQUARTERS

	1-24	25-99	100
21L02 450ns	1.30	1.25	1.15
21L02 250ns	1.59	1.55	1.45
2102	1.25	1.15	1.10
91L02APC	1.75	1.65	1.50
21L11-1	4.10	3.95	3.75
2111-1	3.75	3.65	3.50
21L12-1	4.25	4.10	3.95
2112-1	2.95	2.85	2.65
21L01-1	3.95	3.75	3.65
2101-1	2.90	2.70	2.50
2114-3 300ns	9.95	9.75	9.25
2114-4 450ns	7.95	7.90	5.50
IMS4044	9.95	9.75	9.25
IMS4045	9.95	8.75	8.25
EMM4200A	12.50	11.95	9.95
EMM4402	7.95	7.25	6.25
4101C-8	7.95	7.95	7.25
µPD410 (420ns)	10.95	10.25	9.25
AMD9140/41	10.95	10.25	9.25
AMD9130/31	12.95	11.95	10.25
FSC 460/464 16K CCD Only $18.95 Each			
1101	1.90	1.75	1.25
P2125/93425 (45ns)	7.95	7.35	7.25

WAVEFORM GENERATORS

8038 Function Gen	3.95
MC4024 VCO	2.45
LM566 VCO	1.75
XR2206 Function Generator	5.25

CHARGE COUPLED DEVICES
16K CCD: First time offered Fairchild 460 CCD 16K Memory (now you can experiment with CCD technology at a reasonable price. 17 page Application note supplied with each order. Quantity Limited)
$18.95 each (reg. 43.00)

CRYSTALS
Microprocessor Timebases TV Game

Frequency	Price	Frequency	Price
1.0MHz	$5.95	6.0MHz	$ 3.95
1.8432	4.95	6.144	4.95
2.0MHz	5.95	6.5536	4.95
2.01MHz	2.95	10.0MHz	4.95
2.097152MHz	5.95	13.0MHz	4.95
2.4576MHz	5.45	14.31818	4.95
3.579545MHz	1.50	16.0MHz	4.95
4.0MHz	4.95	18.432MHz	5.95
4.194304MHz	5.95	20.0MHz	4.95
4.9152MHz	5.95	22.1184MHz	5.95
5.0MHz	4.45	27.0MHz	5.95
5.0688	4.95	36.0MHz	5.95
5.7143MHz	9.95	48.0MHz	5.95
		100KHz	12.95

DISPLAYS/OPTO

DL 704/707/CC/CA .300	1.25
FND359 CC .357	.70
FND 500/507/CC/CA .500	1.35
FND 503/510 CC/CA .500	.95
FND 800/807 CC/CA .800	2.50
Mowmar 9 digit bubble	.89
FSC 8024 4 digit CC .800	4.95
HP7340 HEX Display	19.95
TIL 306 5 x 7 Array	8.50
TIL 306 7 seg w/logic	8.95
TIL 308 7 seg w/logic	6.95
TIL 309 7 seg w/logic	5.95
DL 311 HEX Display	12.95
MA 1003 12 volt clock	17.95
MA 1002 4 digit clock module	9.95
MA 1010 4 digit clock module	9.95
NSN 373/374 dual CC/CA .300	2.25
NSN 583/584 dual CC/CA .500	2.60
NSN 783/784 dual CC/CA .700	3.00
4N25 Opto Isolator	1.10
MCT 2 Opto Isolator	.89
4N33 Darlington GOA	1.75
Red Led's .185 Dia	5/1.00
Green/Yellow	4/1.00
HP 5082/7731 7 seg	.90

MONTHLY SPECIALS

14 Pin S/T sockets 1000 pcs.	.10
16 Pin S/T sockets 1000 pcs.	.11
22 Pin S/T sockets 1000 pcs.	.12
FSC460/464 16K CCD's	18.95
AMD 9140CDC Static Ram	10.95
4116/416 16K Dynamic Ram's	8 for 149.95
1488 or 1489 RS232 Driver	1.25
8T97 Buffer	1.25
8130 or 8131	2.45
8833 or 8835	[illegible]
74367 or 74368	[illegible]
75451/52/53	[illegible]
8T26 Bus Driver	[illegible]
78L05 Regulator	[illegible]
MCT2 Optoisolator	[illegible]
2510A Shift Register	[illegible]
2519A Shift Register	[illegible]
MM5554 Shift Register	[illegible]
LM7805N-14 Converter	[illegible]
DS0026CN Clock Driver	[illegible]
DS0026CN Clock Driver	[illegible]
8T20 Bidirect oneshot	[illegible]
1403AN Shift Register	[illegible]
9368	[illegible]

NEW
1978 CATALOGUE
SEND 25¢ POSTAGE

DISCOUNT COMPUTER CORNER
Discount Cash / Computer Credit Card

P. O. BOX 17329 Irvine, California 92713 New Phone (714) 558-8813
TWX: 910-595-1565

Retail Store Open Mon. – Sat.
Located at 1310 "B" E. Edinger, Santa Ana, CA 92705

satisfaction starts here

MEET THE ECONORAM™ FAMILY . . .

These **static** memory kits deliver outstanding performance at prices even the dynamics can't match. What others consider "extras" we consider necessities, such as buffering on all lines, reliable DMA, sockets, gold-plated card fingers . . . and all the other signs of quality that make up an Econoram.

More good news: Our memory kits (except Econoram II™) are now "Unkits". All sockets and bypass caps are pre-soldered into place, turning an otherwise tedious assembly job into a pleasurable, one evening project.

We want to be your memory supplier: and that means offering a superior product at the lowest possible price.

8K ECONORAM II™

Just try to find a more cost-effective S-100 package! Better yet, get 3 kits (24K of memory) for only $375. Add $20 per board for assembled/tested.

$135

16K ECONORAM IV™

Ideal for S-100 buss system builders who need a big chunk of memory that draws less than 2000 mA. Manual write protect for 4K blocks; use with or without phantom line. Add $35 for assembled/tested.

$279

24K ECONORAM VII™

A dense and flexible S-100 board that draws less than 2000 mA. Configured as two 8K and two 4K blocks with independent addressing and protection. Use with or without phantom lines. Add $40 for assembled/tested.

$445

MORE ITEMS . . .

TRS-80 16K CONVERSION KIT

Why settle for the limitations of a TRS-80 4K system? Expand it to 16K with our conversion kit. Includes eight uPD416 1 × 16K dynamic memories and instructions for converting your machine to 16K. You could pay up to $290 elsewhere . . . but our kit is only **$190.00**

CPU POWER SUPPLY

Here is an economical, conservatively rated power supply for small computer systems or digital/analog bench work. Delivers +5V @ 4A with crowbar over-voltage protection; also gives ± 12 @ ½A per side. Adjustable negative bias supply, 5 – 10V @ 10 mA. All in all, if you need a small power supply you can't beat the performance or the price.

$50

11 SLOT MOTHER BOARD "UNKIT" $90

Those who recognize value have made the 10/11 slot Motherboard one of our most popular non-memory kits. Now, we've made it even more desirable by pre-soldering all 11 edge connectors in place to take the tedium out of assembly. Includes our much-copied active termination circuitry that takes the noise, glitches, ringing, and overshoot problems off your buss for optimum data transfer . . . as well as all edge connectors and plenty of bypass capacitors.

18 SLOT MOTHER BOARD KIT $124

Same features as our 11 slot version; includes 18 edge connectors and active termination circuitry.

COMING SOON: A NEW FAMILY OF ECONOROMS™

Don't load and reload often used programs or routines; store them in one of our ROM boards. We're putting final touches on 2 boards; one is 16K, configured as 4 independent 4K blocks with optional power on start, and uses 2708s. Our other board combines both ROM and RAM and uses 2716s. Watch this space for further developments.

GODBOUT

G4

BILL GODBOUT ELECTRONICS
BOX 2355, OAKLAND AIRPORT, CA 94614

TERMS: Please allow up to 5% for shipping; excess refunded. Add $1 handling for orders under $10. Cal res add tax. COD OK with street address for UPS. For VISA® /Mastercharge® orders ($15 min) call our 24 hr. order desk at **(415) 562-0636**. Prices good through cover month of magazine.

600 MHZ. FREQUENCY COUNTER
±0.1 PPM TCXO

OPTO-8000.1

This new instrument has taken a giant step in front of the multitude of counters now available. The Opto-8000.1 boasts a combination of features and specifications not found in units costing several times its price. Accuracy of ±0.1 PPM or better — *Guaranteed* — with a factory-adjusted, sealed TCXO (Temperature Compensated Xtal Oscillator). **Even kits require no adjustment for guaranteed accuracy!** Built-in, selectable-step attenuator, rugged and attractive, black anodized aluminum case (.090" thick aluminum) with tilt bail. 50 Ohm and 1 Megohm inputs, both with amplifier circuits for super sensitivity and both diode/overload protected. Front panel includes "Lead Zero Blanking Control" and a gate period indicator LED. AC and DC power cords with plugs included.

SPECIFICATIONS:
Time Base—TCXO ±0.1 PPM GUARANTEED!
Frequency Range—10 Hz to 600 MHz
Resolution—1 Hz to 60 MHz; 10 Hz to 600 MHz
Decimal Point—Automatic
All IC's socketed (kits and factory-wired)
Display—8 digit LED
Gate Times—1 second and 1/10 second
Selectable Input Attenuation—X1, X10, X100
Input Connectors Type —BNC
Approximate Size—3"h x 7½"w x 6½"d
Approximate Weight—2½ pounds
Cabinet—black anodized aluminum (.090" thickness)
Input Power—9-15 VDC, 115 VAC 50/60 Hz
or internal batteries

OPTO-8000.1 Factory Wired	**$299.95**
OPTO-8000.1K Kit	**$249.95**

ACCESSORIES:
Battery-Pack Option—Internal Ni-Cad Batteries and charging unit **$19.95**
Probes: P-100—DC Probe, may also be used with scope **$13.95**
P-101—LO-Pass Probe, very useful at audio frequencies **$16.95**
P-102—High Impedence Probe, ideal general purpose usage **$16.95**

VHF RF Pick-Up Antenna-Rubber Duck w/BNC #Duck-4H **$12.50**
Right Angle BNC adapter #RA-BNC **$ 2.95**

FC-50 — Opto-8000 Conversion Kits:
Owners of FC-50 counters with #PSL-650 Prescaler can use this kit to convert their units to the Opto-8000 style case, including most of the features.

FC-50 — Opto-8000	**Kit $59.95**
*FC-50 — Opto-8000F	**Factory Update $99.95**
FC-50 — Opto-8000.1 (w/TCXO)	**Kit $109.95**
*FC-50 — Opto-8000.1F	**Factory Update $149.95**

*Units returned for factory update must be completely assembled and operational

OPTOELECTRONICS, INC.
5821 NE 14 Avenue
Ft. Lauderdale, FL 33334
Phones: (305) 771-2050 771-2051
Phone orders accepted 6 days, until 7 p.m. O3

master charge
BANKAMERICARD

TERMS: Orders to U.S. and Canada, add 5% to maximum of $10.00 per order for shipping, handling and insurance. To all other countries, add 10% of total order. Florida residents add 4% state tax. C.O.D. fee: $1.00. Personal checks must clear before merchandise is shipped.

TEKTRONIX 535A with DUAL TRACE PLUG IN (CA).

We have acquired a limited number of TEKTRONIX 535A OSCILLOSCOPES, with the CA (dual trace) plug ins. These scopes have a present market value of about $800.00. For years this scope has been one of ten workhorses of the industry, and it is rare that an individual can find one at an affordable price. Due to limited number available, we suggest telephoning your order in.

All scopes are shipped freight collect.

STOCK NO. XX3K TEKTRONIX 535A Oscilloscope and CA (dual trace) Plug in, $495.00

FORM FEED MECHANISM

Again DELTA brings you the unusual. This FORM FEED MECHANISM is designed to be used with any printer, is completely adjustable in width and speed, which is controlled by your printer. It was made in West Germany, and was made to sell for $450.00. We will sell the limited quantity we have for $99.00. This is a device that you may never again see advertised.

STOCK NO. XXX4K FORM FEED MECHANISM $99.00

TV HOCKEY and SOCCER GAME plus VIDEOCUBE

For some time we have been selling our VIDEOCUBE, the TV interface between computers, cameras, etc., and your TV set. We sold them in kit form, and to date we have sold over 3,000. We sold the complete VIDEOCUBE, when available, for $16.95.

We now have a limited number of Hockey-Soccer Video games, complete with VIDEOCUBE interface, at a new price of $18.95; previously sold for $27.50 in kit form. This gives you two games, at two levels of skill, with LED readouts for scorekeeping. This game sold in Boston department stores for $79.50.

STOCK NO. 5495K Complete VIDEO GAME & VIDEOCUBE $18.95 ea. 2/$35.00

NEW TRANSFORMERS

TRANSFORMER 1. Primary 115V, Secondary 1, 30 V @ 2 A. Secondary 2, 16.5 V @ 1.2 A. Secondary 3, 16 V @ 3.5 A. Secondary 4,9.5 V @ 3.5 A.

STOCK NO. 6677R Wt. 10 lbs. $10.95 ea. 2/20.00

TRANSFORMER 2. Primary, 115 V. Secondary 1, 12 V. @ 5 A., Secondary 2, 24 V. @ 9 A., Secondary 3 14 V. @ 20 A. Secondary 4, 125 V. @ 1.5 A.

STOCK NO. 6675K Wt. 16 lbs. $18.95 ea. 2/36.00

TRANSFORMER 3. Primary 115 V. Secondary 1, 12.6 V ct @ 1 A. Secondary 2, 33 V tapped at 13.1 V @ 1 Amp. Secondary 3, 140 V. @ 100 ma.

STOCK NO. 6772K 2 lbs. 4 1/16" x 2 1/16" x 2 5/8" $2.95 ea. 2/5.00

Please include sufficient postage, unless otherwise noted. Excess will be refunded. Send for our NEW CATALOG 19, 120 pages of computer and electronic bargains.

MAIL ORDER

DELTA ELECTRONICS
P.O. Box 2, 7 Oakland St.
Amesbury, Mass. 01913

Over the counter salesroom

ATLANTA GA
DELTA ELECTRONIC HOBBIES
5151 Buford Hwy.
Doraville, Atlanta, Ga.

DELTA ELECTRONIC WAREHOUSE OUTLET 590 Commonweath Ave, Boston, Mass, is no longer affiliated with DELTA ELECTRONICS D13

S.D. COMPUTER PRODUCTS

AN EMPIRE IND. CO.

P.O. BOX 28810K DALLAS, TEXAS 75228

EXPANDABLE EPROM BOARD

16K OR 32K EPROM $49.95 W/OUT EPROM

Allows You to Use Either 2708's For 16# of Eprom or 2716's For 32K of Eprom.

KIT FEATURES:

1. All Address Lines & Data Buffered
2. Quality Plated through P.C. Board Including Solder Mask and Silk Screen
3. Selectable Unit States
4. On Board Regulation Provided
5. All Sockets Provided W/Board

WE CAN SUPPLY 450Ns 2708's AT $11.95 WHEN PURCHASED WITH BOARD.

EXPANDORAM
THE ULTIMATE RAM BOARD
32K FOR $475.00

32K MODEL	
8K	$151.00
16K	259.00
24K	367.00
32K	475.00

64K MODEL	
16K	$281.00
32K	519.00
48K	757.00
64K	995.00

THE 32K VERSION USES THE MOSTEK MK4115 RAM AND HAS 8K BOUNDARIES AND PROTECTION & UTILIZES DIP SWITCHES. P. C. BOARD COMES WITH SOCKETS FOR 32K OPERATION

BUY A S100 COMPATIBLE RAM BOARD AND UPGRADE THE SAME BOARD TO A MAXIMUM OF 65K MEMORY IN STEPS OF EITHER 8K OR 16K AT YOUR OPTION BY MERELY PURCHASING MORE RAM CHIPS FROM SD COMPUTER PRODUCTS.

THE 64K VERSION USES THE MOSTEK MK4116 RAM AND HAS 16K BOUNDARIES AND PROTECTION & UTILIZES DIP SWITCHES. P. C. BOARD COMES WITH SOCKETS FOR 64K OPERATION

LOOK AT THE FEATURES WE HAVE BUILT INTO THE EXPANDORAM!

- MEMORY ACCESS TIME IS 375 ns
- MEMORY CYCLE TIME IS 500 ns
- POWER REQUIREMENTS ARE:
  - 8 VDC 400 MA DC
  - 18 VDC 400 MA DC
  - —18 VDC 30 MA DC
- ON BOARD INVISIBLE REFRESH
- NO WAIT STATES REQUIRED
- NO CYCLE STEALING NEEDED
- ON BOARD REGULATION
- CONTROL, DATA & ADDRESS INPUTS UTILIZE LOW POWER SCHOTTKY DEVICES
- DESIGNED TO WORK WITH Z-80, 8080, 8085, CPU's

ADD $50.00 TO ABOVE PRICES FOR FULLY ASSEMBLED AND TESTED BOARDS

Low Cost Cassette Interface Kit

$19.95

Features: Play and record K.C. Standard 2400/1200 Hz tapes, 300 Baud, TTL I/O Compatible, Phase Lock Loop, Both 22 Pin Connector and 8 Pin Molex Connector. Comes partially assembled. Oscillator and phase lock loop pre-tuned to K.C. Standard. Selector switch sends cassette data or auxiliary input data to microprocessor. LED indicates logic 1 level.

4K LOW POWER RAM KIT
The Whole Works - $79.95

Full Buffered - on board regulated - reduced power consumption utilizing low power 21L02-1 500ns RAMS - Sockets provided for all IC's. Quality plated through PC board.

*Add $10.00 for 250ns RAM operation.

8K LOW POWER RAM
$159.95

FULLY ASSEMBLED AND TESTED, NOT A KIT.

Imsai — Altair — S-100 Buss compatible, uses low power static 21L02-500ns fully buffered on board regulated, quality plated through PC board, including solder mask. 8 pos. dip switches for address select.

*Add $30.00 for 250ns RAM operation

Z-80 CPU BOARD KIT
Complete Kit $139.

CHECK THE ADVANCED FEATURES OF OUR Z-80 CPU BOARD: Expanded set of 158 instructions, 8080A software capability, operation from a single 5VDC power supply; always stops on an M1 state, true sync generated on card (a real plus feature!), dynamic refresh and NMI available, either 2MHZ or 4MHZ operation, quality double sided plated through PC board; parts plus sockets provided for all IC's. *Add $10. extra for Z-80A chip which allows 4MHZ operation.

NEW FROM S.D.
"VERSAFLOPPY"™ KIT

THE VERSATILE FLOPPY DISK CONTROLLER

ONLY $149.00

FEATURES: IBM 3740 Soft Sectored Compatible. S-100 BUS Compatible for Z-80 or 8080. Controls up to 4 Drives (single or double sided). Directly controls the following drives:

1. Shugart SA400/450 Mini Floppy
2. Shugart SA800/850 Standard Floppy.
3. PERSCI 70 and 277.
4. MFE 700/750.
5. CDC 9404/9406.

34 Pin Connector for Mini Floppy 50 Pin Connector for Standard Floppy. Operates with modified CP/M operating system and C-Basic Compiler. The new "Versafloppy" from S.D. Computer Products provides complete control for many of the available Floppy Disk Drives, Both Mini and Full Size. FD1771B-1 Single Density Controller Chip. Listings for Control Software are included in price.

FD 1771B-1 CHIP ALONE $39.95

Z80 STARTER KIT

LEARN COMPUTERS FROM THE START!

SIMPLE, STEP BY STEP LEARNING. CONSTRUCTION, PROGRAMMING, OPERATION, MEMORIES, INTERFACING, COMPUTING, AND CONTROLLING WITH AUDIO CASSETTE INTERFACE CAPABILITIES.

Complete Kit includes: Key board and Display; Z80 Central Processing Unit; Instructions; Operation Manual; Learning Guides.

Features: Powerful Z80 CPU with 158 instructions * 1024 Bytes (Expandable to 2048 Bytes ON BOARD) of RAM * 2 Bi-Directional Input/Output Ports with Handshaking * Kansas City Standard Audio Cassette Interface for Program Storage * Hexadecimal Keyboard and Display * Wirewrap area for custom circuitry * S-100 Connector on board for Memory and I/O Expansion * 2716/2758 PROM Programmer * "Z-BUG" Monitor ROM (Including: Memory, Port and Register Examine and Change Commands; Breakpoints; Single Step Capability; Audio Tape Load and Dump; Execute user program Commands.)

Many more unique features. The best computer educational kit on the market . . . **the complete computer and educational package for only $199.00.** (Available June 1978).

INTRODUCING THE SBC-100

(The Z-80 Based, S-100 Single Board Computer)

$349.00

FEATURES:

- No Front Panel Needed
- Z-80 CPU (2 or 4 MHZ)
- 1K RAM
- 4 ROM/PROM Sockets for 4K/8K of Memory
- SYNCHRONOUS/ASYNCHRONOUS Serial I/O with RS-232 and Current Loop Interface and Software
- Programmable Baud Rate
- Parallel Input Port
- Parallel Output Port
- 4 Channel Timer/Counter
- 4 Vectored Interrupts

O.E.M. SPECIAL

ASK ABOUT SPECIAL O.E.M. DISCOUNTS ON THE S.D. "COMPATIBLE SET."

SBC-100 — SINGLE BOARD COMPUTER	$349.00
VERSAFLOPPY™ — FLOPPY DISK CONTROLLER	$149.00
EXPANDORAM — 32K RAM	$475.00

EACH KIT IF PURCHASED SEPARATELY TOTAL $973.00

ORDER ALL 3 KITS TOGETHER FOR

$899.00

This Powerful Threesome Operates Together to Form A Complete Computer for Your System.

Z-80 Programming Manual — IN DEPTH DETAIL OF THE Z-80 CPU MICRO-COMPUTER — **S. D. SALES SPECIAL $9.95**

RAMS

21L02 - 500NS	8/11.50
21L02 - 250NS	8/15.95
2114 — 4K	14.95
1101A — 256	8/$4.00
1103 — 1K	.35
MK 4115 - 8K	15.45
745 200 - 256	3.95

CPU's

Z—80 includes manual	29.95
Z—80A includes manual	34.95
8080A CPU 8 BIT	11.95
8008 CPU 8 BIT	6.95

PROMS

1702A - 1K - 1.5us	3.95 or 10/35.
2708 - 8K - 450ns	14.95
5204 - 4K	7.95
82S129 — 1K	2.50
2708U 8K signetics 650ns	9.95

COUNTER CHIPS

MK50397 6 Digit elapsed timer	8.95
MK50250 Alarm clock	4.99
MK50380 Alarm chip	2.95
MK50395 6 digit up/dn. count.	12.95
MK5002 4 digit counter	8.95
MK5021-Cal. chip sq. root	2.50

S.D. NOW HAS SOFTWARE FOR IT'S CUSTOMERS

CP/M DISK OPERATING SYSTEM . . . $99.95

CP/M is a powerful disk operating system which has become an industry standard. It is compatible with several disk based FORTRAN and BASICS. This package includes a CP/M diskette (mini or full size) adapted for S.D.'s SBC-100/VERSAFLOPPY EXPANDORAM board set. Complete documentation is included. (CP/M is a registered trademark of Digital Research Corp., Pacific Grove, CA.

Z-80 DISK BASED ASSEMBLER $69.95

Runs on ANY CP/M based disk system. Assembles the official Zilog-Mostek Mnemonics. Contains extensive set of pseudo-ops. Available on mini or full size diskette.

VERSAFLOPPY™ CONTROL FIRMWARE . . . $24.95

Provides control for VERSAFLOPPY and boots up CP/M. This runs on Z-80, 8080 or 8085 based computers. Available in 2708 or 2758 prom.

SD MONITOR $49.95

Powerful monitor for SBC-100 single board computers. Includes all VERSAFLOPPY control firmware. Comes in 2716 prom. Available in 4-6 weeks.

VERSAFLOPPY DIAGNOSTIC PROGRAM . . . $24.95

Provides routines which are helpful in checking out a disk based system. Available in 2708 or 2758 prom.

SUPER FLOPPY SPECIAL

S. D. SALES' VERSAFLOPPY S-100 CONTROLLER BOARD PLUS SHUGART SA 400 FLOPPY DISK DRIVE INCLUDING CABLE FOR ONLY

$479.00

MICROPROCESSOR CHIPS

8212 - I/O port	3.50
8214 — P.I.C.	12.95
8216 — Non Invert Bus	4.95
8224 — Clock Gen.	4.95
8226 — Invert Bus	3.95
PIO for Z—80	14.95
CTC for Z—80	14.95
8228 Sys. Controller	8.20
8251 Prog. comm. interfact	10.95
8255 prog. prep. interface	13.50
8820 Dual Line Recr	1.75
8830 Dual Line Dr.	1.75
2513 Char. Gen.	7.50
8838 Quad Bus. Recvr.	2.00
74LS138N — 1/8 decoder	.99
8T97-Hex Tri-State Buffer	1.25
1488/1489 RS232	1.50
TR 1602B Uart	3.95
TR 1863 Uart	8.50
FD 1771B-1	39.95

CMOS

4001	19	4029	99
4002	19	4042	69
4011	19	4047	1.50
4013	32	4049	35
4016	32	4069	23
4017	95	4071	19
4020	97	4076	97
4022	97	14518	1.10
4024	75	14528	85
4027	39	14529	85

CALL IN YOUR BANKAMERICARD (VISA) OR MASTER CHARGE ORDER IN ON OUR CONTINENTAL TOLL FREE WATTS LINE:

1—800—527—3460

Texas Residents Call Collect:

214/271—0022

DEALER INQUIRIES INVITED!

(All prices subject to change without prior notice.)

NO COD's. TEXAS RESIDENTS ADD 5% SALES TAX. ADD 5% OF ORDER FOR POSTAGE & HANDLING. ORDERS UNDER $10. ADD 75c HANDLING. FOREIGN ORDERS — U. S. FUNDS ONLY!

FAIRCHILD FND 507 and HP 5082-7750

½" Common Anode READOUTS

STOCK NO. 5616F is a 3½ digit assembly, (3 digits and ± 1). The assembly has been removed from equipment. The 4 readouts are guaranteed. You may also find an SN7447 driver/decoder and segment dropping resistors. Transistors have been removed.

STOCK NO. 5616F	3½ digit display, (4 readouts)	$2.50	3/6.00

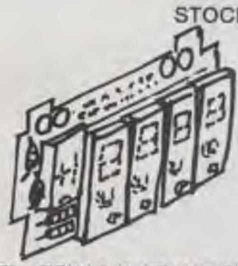
STOCK NO. 5620F is a 4½ digit HEWLETT-PACKARD 5082-7750 common anode display, ½" high. This assembly has been removed from equipment, and the 5 readouts are guaranteed. You might also find on the assembly segment dropping resistors and IC drivers. Transistors have been removed.

STOCK NO. 5620F	4½ digit display, (5 readouts)	$3.50	3/9.00

RCA NUMITRON 2½ and 3½ DIGIT ASSEMBLIES

These units consist of 2½ and 3½ digit assemblies of RCA NUMITRON 7 segment incandescent readouts, plus a ± readout. These readouts lend themselves to use of many different color filters, so that almost any color is obtainable. These assemblies are out of equipment, and most contain several logic chips, and a trimpot for adjusting brightness. Many of the 2½ digit chips contain 3 SN7475s and 2 SN74472. The 3½ digit assemblies may contain 3 SN7447s, 2 SN74174s and a SN7475.

STOCK NO. 5622F	2½ digit RCA NUMATRON ASSEMBLY	$3.00	2/5.00
STOCK NO. 5621F	3½ digit RCA NUMITRON ASSEMBLY	$4.50	2/8.00



TEXAS INSTRUMENT TIL 307
7 SEGMENT READOUT WITH LOGIC
(COUNTER—LATCH—DECODER).

The TEXAS INSTRUMENT TIL-307 is a 7 segment readout, with all the logic necessary for counters built into the chip. This simplifies devices, and makes for small packages in instrumentation. The overall dimensions of the readout are 3/8" x 1" and the digit is .270". The device mounts in a 16 pin DIP socket, and all that is needed is the input signal. These devices cost $11.71 each at the distributors. The ones we have are out of equipment, but are 100% guaranteed. We have 2 types, 3½ digits, and 4½ digits. The 3½ digits contain 3 TIL 307s, and the 4½ digits contain 4 TIL 307s. We provide data with these readouts. The ½ digit in each case is a ± 1.

STOCK NO. 5614F	3½ digit TIL 307 with logic & data	$10.00	2/19.00
STOCK NO. 5619F	4½ digit TIL 307 with logic & data	$12.50	2/24.00

OVENAIRE ULTRA PRECISION CRYSTAL OSCILLATOR

Your computer is only as good as its clock. we have been fortunate in acquiring a lot of OVENAIRE precision crystal oscillators, Model OSC 67-11-A-3. The output frequency of these oscillators is 3.840 MHz. The frequency readily divides into many useable frequencies with the use of standard SN7400 series ICs. Among the many frequencies are 640 KHz, 60 KHz, 32 KHz, 20 KHz, 10 KHz, 6 KHz, 1 KHz, 600 Hz, 100 Hz, 60 Hz, 50 Hz and many more. We provide data showing the ICs needed to get these frequencies. The oscillator is precise to 2 parts per million, and is adjustable to even greater precision. Ideal for computers, frequency standards, clocks, etc. This oscillator is a current production item, and the one piece price at the factory is $134.50. In lots of 100 the price is $49.80, so our price of $14.95 each is a fantastic bargain. 1 5/8" x 2" x 5/8". PC mount. Voltages required are 5 Vdc and 12 Vdc. Output is TTL compatible 5 Vdc. Sketch at left shows the complete unit, and an inside view.

STOCK NO. 5592K	Ovenaire Precision Crystal Oscillator	$14.95 ea.	2/28.00

Enclose sufficient postage. Excess will be refunded. Send for NEW CATALOG 19, 120 pags of computer and electronic bargains.

MAIL ORDER
DELTA ELECTRONICS
P.O. Box 2, 7 Oakland St.
Amesbury, Mass. 01913

Over the counter salesroom
ATLANTA GA
DELTA ELECTRONIC HOBBIES
5151 Buford Hwy.
Doraville, Atlanta, Ga.

DELTA ELECTRONIC WAREHOUSE OUTLET
590 Commonweath Ave, Boston, Mass. is no longer affiliated with DELTA ELECTRONICS

D13

4804 STATIC, TTL IN/OUT 1024x4 N-MOS RAM

GENERAL DESCRIPTION

Part Number 4804 is a 4K semiconductor random access memory organized as 1024 4-bit words. It is fully static and needs no clock or refresh pulses. It requires a single +5 volt power supply and is fully TTL compatible on input and output lines. The 4804 is packaged in a convenient 18 pin dual-in-line package.

FEATURES

- Single +5V Power Supply
- 1Kx4 Organization
- Replaces 4 1024x1 Static RAMs
- Completely Static—No Clocks or Refresh
- 18 Pin Package
- Access/Cycle 600nsec max
- 250 mw Typical Operating Power
- Common I/O Bus
- TTL Compatible I/O
- Three State Outputs

TRUTH TABLE

CE	R/W	DI/DO	STATUS	MODE
H	Don't Care	High Z	Deselect	Standby
L	H	Data	Selected	READ
L	L	L	Selected	Write 0
L	L	H	Selected	Write 1

WRITE CYCLE—AC CHARACTERISTICS

PARAMETER	SYMBOL	4804 MIN	MAX
Write Cycle Time	Twc	600	
Address To Write Time	Taw	100	
Write Pulse Width	Twp	500	
Write Recovery Time	Twr	0	
Data Set Up Time	Tdw	350	
Data Hold Time	Tdh	0	
Output Disable From Write or Chip Enable	Totw		150

READ CYCLE—AC CHARACTERISTICS

PARAMETER	SYMBOL	4804 MIN	MAX
Read Cycle Time	Trc	600	
Access Time	Ta		600
Chip Enable to Output Enable	Tco		200
Data Valid After Address	Toh	150	
Previous Data Valid After Chip De-Select	Totd	25	

$8.95 8/$60.00 16/$100.00

INTEGRATED TONE RECEIVER
MK5102(N)-5

FEATURES

- ☐ Detects all 16 standard DTMF digits
- ☐ Requires minimum external parts count for minimum system cost
- ☐ Uses inexpensive 3.579545 MHz crystal for reference
- ☐ Digital counter detection with period averaging insures minimum false response
- ☐ 16-pin package for high system density
- ☐ Single supply 5 Volts ± 10%
- ☐ Output in either 4-bit binary code or dual 2-bit row/column code
- ☐ Latched outputs

DESCRIPTION

The MK5102 is a monolithic integrated circuit fabricated using the complementary-symmetry MOS (CMOS) process. Using an inexpensive 3.579545 MHz television colorburst crystal for reference, the MK5102 detects and decodes the 8 standard DTMF frequencies used in telephone dialing. The requirement of only a single supply and its construction in a 16-pin package make the MK5102 ideal for applications requiring minimum size and external parts count.

DETECTION FREQUENCY

Low Group f_o	High Group f_o
Row 1 = 697 Hz	Column 1 = 1209 Hz
Row 2 = 770 Hz	Column 2 = 1336 Hz
Row 3 = 852 Hz	Column 3 = 1477 Hz
Row 4 = 941 Hz	Column 4 = 1633 Hz

MK5102N-5..............................$34.95
Specs..50
600 Ohm to 600 Ohm C.T. transformer.........$1.95
Colorburst crystal for above...................$1.75

TRI-TEK

T1

10 AMP REGULATORS
78P05

GENERAL DESCRIPTION— The uA78P05 is a 3-terminal positive 5V hybrid regulator capable of delivering 10 Amps! This device is virtually blowout proof and contains all the protection features inherent in monolithic regulators such as internal short-ciruit current limiting and thermal-overload protection. The uA78P05 is packaged in a hermetically sealed TO-3 providing 50W at 25° C case. The hybrid consists of a monolithic control chip driving a rugged Mesa transistor. The high output current is achieved through new design technique without sacrificing the regulation characteristics of the controlling elements. The same process is employed in the construction of the 10A regulator to provide the same high reliability obtained in the uA78H05 5A regulator.

- 10 A OUTPUT CURRENT
- INTERNAL THERMAL-OVERLOAD PROTECTION
- INTERNAL SHORT-CIRCUIT CURRENT LIMIT
- LOW DROP-OUT VOLTAGE 2.2 V AT 10 A
- 50 W POWER DISSIPATION
- PIN-FOR-PIN COMPATIBLE WITH THE μA78H05, μA78H05A AND SH323
- STEEL TO-3 PACKAGE

CONNECTION DIAGRAMS
TO-3 PACKAGE
(TOP VIEW)

78P05SC.......................................$12.95
Specs..60

MULTI-CHANNEL 8 BIT A/D CONVERTER

Fairchilds' new 6 channel analog-to digital converter has a lot going for it. Full scale correction capabilities, ratiometric conversion and wide input dynamic range.
Micro-processor compatible, it combines the multiplexer, decoder and sample-and -hold functions with converter to save board space and eliminate external parts. It provides 8 bit, ±½ LSB conversion in 300 uSec featuring auto-zero and dynamic range all the way to ground.
UA9708 in 16 pin plastic DIP.......................$7.95!

DIXON
THE THIRD HAND

- This versatile tool has proven its worth through generations of use by professional craftsmen.
- The Third Hand holds work in any position freeing both hands to perform other vital functions.

CROSSLOCK TWEEZER
Dixon #H-500

- Tweezer exerts 1½ lb. tension on blunt ends
- Double ball joint allows positioning of tweezer at any angle
- Heavy iron base may be held in vise or clamped to bench

- FREES HANDS

Dixon Third Hand...$7.95

4801 STATIC, TTL IN/OUT 4096x1 N-MOS RAM

GENERAL DESCRIPTION

Part Number 4801 is a 4K semiconductor random access memory organized as 4096 1-bit words. It is fully static and needs no clock or refresh pulses. It requires a single +5 volt power supply and is fully TTL compatible on input and output lines. The 4801 is packaged in a convenient 18 pin dual-in-line package.

FEATURES

- Single +5V Power Supply
- 4Kx1 Organization
- Replaces 4 1024x1 Static RAMs
- Completely Static—No Clocks or Refresh
- 18 Pin Package
- Access/Cycle Times 600 nsec max
- 250 mw Typical Operating Power
- Separate Data In and Data Out
- TTL Compatible I/O
- Three State Outputs
- Data Bus Compatible I/O Function

TRUTH TABLE

CE	R/W	DI	DO	STATUS	MODE
H	Don't Care	Don't Care	High Z	Deselect	Standby
L	H	Don't Care	Data	Selected	READ
L	L	L	High Z	Selected	Write 0
L	L	H	High Z	Selected	Write 1

WRITE CYCLE—AC CHARACTERISTICS

PARAMETER	SYMBOL	4801 MIN	MAX
Write Cycle Time	Twc	600	
Address To Write Time	Taw	100	
Write Pulse Switch	Twp	500	
Write Recovery Time	Twr	0	
Data Set Up Time	Tdw	350	
Data Hold Time	Tdh	0	
Output Disable From Write or Chip Enable	Totw		150

READ CYCLE—AC CHARACTERISTICS

PARAMETER	SYMBOL	4801 MIN	MAX
Read Cycle Time	Trc	600	
Access Time	Ta		600
Chip Enable to Output Enable	Tco		200
Data Valid After Address	Toh	150	
Previous Data Valid After Chip De-Select	Totd	25	

$8.95 8/$60.00 16/$100.00

VOLTAGE REGULATORS

7805-06-08-12-15-24 TO220	95¢ 5/$4.50
78LO5A-12-15 4% 100 mA TO-92 Plastic	50¢
78HO5KC 5V 5A TO-3	8.45
78H12KC 12V 5A TO-3	9.15
78H15KC 15V 5A TO-3	9.15
Lm317K 1.5A *Adjustable* TO-3	4.99
Lm317T 1.5A *Adjustable* TO-220	3.99
Lm317MP .5A *Adjustable* TO-202	13.95
TL430C Adjustable Zener-*Think About It*	1.50
TL497C Switching Reg. & Inductor	9.50
RCA CA 3085 100 mA *Adjustable*	.60
Signetics 2504TA 1024 bit S.R. memory (1404A)...	.50
MCM 6571P Character Generator	9.95
MCM6571AP Character Generator	9.95
MC14409P Telephone Rotary Pulser	10.98
MC14419P Touch Pad Converter for 14409	4.25
MC14411P Baud Rate Generator	11.98
MC14412VP CMOS Modem Chip	16.95
MM57109N Number Cruncher Micro	18.95
74C915 7 Segment to BCD Converter	2.99
74C922 16 Key Keyboard Encoder	6.35
74C923 20 key Keyboard Encoder	6.45
74C925 4 Decade Counter w/latches	12.00
74C926 4 Decade Counter w/carry	12.00
74C935-1 3½ Digit DVM CMOS Chip	16.98

Jumper Kits for .025 Square Posts.. All material for making jumpers for quick circuit changes and prototyping. Use for breadboarding, trouble shooting, field modifications. Fits standard IC socket wire/wrap posts. Excellent wiping action on gold plated box contacts.
Kit contain 10 box contacts, heat shrinkable sleeving, and 5 feet of wire plus instruction sheet.
JCK-5101....(5 double jumpers)$2.75, 4 kits/$10.00

DEALER PRICING AVAILABLE UPON REQUEST

TRI-TEK
7808 North 27th Avenue
Phoenix, Arizona 85021
(602) 995-9352

- Please give street address for UPS shipping when possible.
- C.O.D. NO parcel post C.O.D. UPS C.O.D. Add $1.10 to order.
- Any correspondence not connected with your order, please use separate sheet and include S.A.S.E. for reply.
- Orders less than $10 ($15 foreign) please add $1 handling.
- Prices are subject to change without notice.
- Any refunds will be by check, not credit vouchers.
- Terms: Check, money order, credit card. Net 30 days to rated firms, schools and government agencies.
- If we should be temporarily out of stock on an item, it will be placed on back order. If we cannot ship in 30 days, you will be notified of the expected shipping date and furnished with a postage paid card with which to cancel your order if desired.
- We pay surface shipping only in USA, Canada and Mexico.
- For premium shipping (first class, special handling, etc.) add extra. Excess will be refunded.
- Foreign orders (except Canada and Mexico) estimate and add shipping. Excess will be refunded.

Charge card telephone orders ($20 min.) will be accepted 9-5:30 P.M. except weekends. Telephone 995-9352. No collect calls please.

(215) 279-7988

RNB ENTERPRISES INCORPORATED

P.O. Box 254
King of Prussia, Pa. 19406

DEAR CUSTOMERS AND FUTURE CUSTOMERS:

If you are or are considering to be an owner of either a VIM-1 or KIM-1 microcomputer system, we are announcing a full set of expansion boards *specifically designed for you. No longer* will you have the aggravation and cost of finding out which of the S-100 Bus Boards *might* work (a good number of them will not) and what modifications may be needed to make them a part of your system. The S-100 Bus Boards were designed to meet the 8080 microprocessor's timing requirements which are quite different from that of the 6502 microprocessor. The boards described on the next page are our initial offering in expansion boards designed specifically for the VIM-1 and the KIM-1 systems. There are more items such as an intelligent floppy disk system, I/0 board, wire-wrap board, etc. that are now in our design engineering group.

Have you ever had or been concerned about the misfortune of losing many hours of work because of a failure. And then, to make things worse, find yourself without your system for weeks while it is being repaired. Well, all products that we manufacture are designed to meet or exceed industrial standards. All components are first quality and meet full manufacturer's specifications. All this and an extended burn-in is done to reduce the normal percentage of field failures by up to 75%. To you, this means the chance of inconvenience and lost time due to a failure is very rare; but, if it should happen, we guarantee a turn-around time of less than forty-eight hours for repair.

Our money-back guarantee: If, for any reason you wish to return any board that you have purchased directly from us within ten (10) days after receipt, complete, in original condition, and in original shipping carton; we will give you a complete credit or refund less a $10.00 restocking charge per board.

In order to serve you more efficiently, we are moving our corporate offices to Phoenix, Arizona. This move will combine our offices with our existing manufacturing facilities. This move will greatly increase our internal coordination and allow us to serve you even faster. *EFFECTIVE 1 SEPTEMBER, 1978,* our new address and phone number will be:

2967 W. Fairmount Ave., Phoenix, Arizona 85017 (602) 265-7564

If you are a dealer, we are looking for a few more qualified distributors for our quality products. Please send us a letter telling us about your operation, and, if you qualify, we will send you our dealer price list.

To facilitate your ordering, we have provided a handy order blank below:

Thank you.

Respectfully,
Raymond N Bennett
RNB ENTERPRISES, INC.
Raymond N. Bennett
President

THE EXCLUSIVE ALTERNATIVE FOR QUALITY

Master Charge VISA

Enclosed: ☐ check ☐ M.O. Charge: ☐ VISA ☐ Master Charge All Items Shipped Prepaid Ups

Card # ____________ Interbank # ____________

Expiration date: ____________ Signature: ____________

QTY	ITEM	PRICE	QTY	ITEM	PRICE
	SUB TOTAL			SUB TOTAL	

TOTAL AMOUNT$ ____________

PA. AND AZ. RESIDENT MUST ADD SALES TAX$ ____________

TOTAL AMOUNT OF ORDER$ ____________

NAME: ____________

ADDRESS: ____________

CITY: ____________ STATE: ________ ZIP: ________

7400 TTL

[illegible]

20% Discount 100 pcs combined order 25% - 1000 pcs combined order

C/MOS

[illegible]

74C00

[illegible]

LINEAR

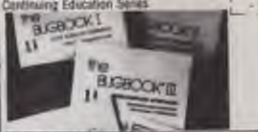
[illegible]

74LS00 TTL

[illegible]

BUGBOOK®

Continuing Education Series



THE 555 TIMER APPLICATIONS SOURCEBOOK WITH EXPERIMENTS by Howard M. Berlin W3HB — $6.95

This book shows you what the 555 timer is and how to use it. Included are over 100 various design techniques, equations and graphs to create "ready-to-go" timers, generators, power supplies, measurement and control circuits, party games, circuits for the home and automobile, photography, music and Amateur Radio.

INSTRUCTOR'S MANUAL — $3.00

Necessary for instruction of Bugbook I and II. Answers questions regarding experiments, suggestions for further reading, philosophy of authors approach to digital electronics. A must for self-learning individuals.

OP AMP MANUAL by Howard M. Berlin W3HB — $9.00

An experiment guide to application of operational amplifiers. Over 25 experiments on all phases of Op Amps.

DBUG — $8.00

8080 interpretive debugger. A program for entering, debugging and storing assembly language programs.

BUGBOOKS I and II by Peter R. Rony, David G. Larsen, WB4HYJ — $17.00 per set

Sold as a set these two books outline over 90 experiments designed to teach the reader all he will need to know about TTL logic chips to use them in conjunction with microcomputer systems. You'll learn about the basic concepts of digital electronics including gates, flip-flops, latches, buses, decoders, multiplexers, demultiplexers, LED displays, RAM's, ROM's, and much, much more.

BUGBOOK IIa by Peter R. Rony, David G. Larsen, WB4HYJ — $5.00

This volume will introduce you to the fabulous UART chip — that all important interface between data terminals, etc., and your microcomputer. It also covers current loops, and the RS 232C interface standard. Particularly recommended for any RTTY enthusiast.

BUGBOOK III by Peter R. Rony, David G. Larsen, WB4HYJ, Jonathan A. Titus — $15.00

Here is the book that puts it all together. Besides having much valuable text there are a series of experiments in which the reader completely explores the 8080 chip pin by pin and introduces you to the Mark 80 microcomputer. A unique easily interfaced system. It is recommended that you have the background on the BUGBOOKS I & II before proceeding with BUGBOOK III.

BUGBOOK V and VI by David G. Larsen, Peter R. Rony, Jonathan A. Titus — $19.00 per set

Experiments in digital electronics, 8080A microcomputer programming and 8080A microcomputer interfacing. An integrated approach to self-instructed basic digital electronics, breadboarding and 8080A interfacing/programming. Bugbook VI integrates the digital concepts of Bugbook V into a treatment of 8080A microcomputer programming and interfacing. Over a laboratory experiments included with each book.

CMOS-III — DESIGNERS PRIMER AND HANDBOOK New expanded version — $9.50

Starts at basic structure of CMOS devices through integration into MSI.

COMPLETE MANUAL FOR DIGITAL CLOCKS by John Weiss and John Brooks — $3.95

Familiarizes technician or hobbyist with basic theories behind digital clocks. Includes trouble-shooting guides, basic characteristics of clocks, soldering techniques, clock component data sheets and construction tips.

DISCRETE LEDS

XC209	Red	5/$1
XC209	Green	4/$1
XC209	Yellow	4/$1
XC111	Red	5/$1
XC111	Green	4/$1
XC111	Yellow	4/$1

[illegible]

DISPLAY LEDS

[illegible]

RCA LINEAR

CA3013	2.15	CA3082	2.00
CA3023	2.56	CA3083	1.60
CA3035	2.48	CA3086	.85
CA3039	1.35	CA3089	3.75
CA3046	1.30	CA3130	1.39
CA3059	3.25	CA3140	1.25
CA3060	3.25	CA3160	1.25
CA3080	.85	CA3401	.49
CA3081	2.00	CA3600	3.50

CALCULATOR CHIPS AND DRIVER

FCM3817	$ 5.00
MM5725	2.95
MM5736	1.95
MM5738	2.95
DM8864	2.00
DM8865	1.00
DM8887	.75
DM8889	.75
5030	7.95

CLOCK CHIPS

[illegible]

8374 - 7-segment LED driver common anode LEDs $ .99

IC SOLDERTAIL — LOW PROFILE (TIN) SOCKETS

SOLDERTAIL STANDARD (TIN)

SOLDERTAIL STANDARD (GOLD)

WIRE WRAP SOCKETS (GOLD) LEVEL #3

[illegible]

50 PCS. RESISTOR ASSORTMENTS $1.75 PER ASST.

ASST. 1	5 ea.	10 OHM / 27 OHM	12 OHM / 33 OHM	15 OHM / 39 OHM	18 OHM / 47 OHM	22 OHM / 56 OHM	1/4 WATT 5%	50 PCS.
ASST. 2	5 ea.	68 OHM / 180 OHM	82 OHM / 220 OHM	100 OHM / 270 OHM	120 OHM / 330 OHM	150 OHM / 390 OHM	1/4 WATT 5%	50 PCS.
ASST. 3	5 ea.	470 OHM / 1.2K	560 OHM / 1.5K	680 OHM / 1.8K	820 OHM / 2.2K	1K / 2.7K	1/4 WATT 5%	50 PCS.
ASST. 4	5 ea.	3.3K / 8.2K	3.9K / 10K	4.7K / 12K	5.6K / 15K	6.8K / 18K	1/4 WATT 5%	50 PCS.
ASST. 5	5 ea.	22K / 56K	27K / 68K	33K / 82K	39K / 100K	47K / 120K	1/4 WATT 5%	50 PCS.
ASST. 6	5 ea.	150K / 390K	180K / 470K	220K / 560K	270K / 680K	330K / 820K	1/4 WATT 5%	50 PCS.
ASST. 7	5 ea.	1M / 2.7M	1.2M / 3.3M	1.5M / 3.9M	1.8M / 4.7M	2.2M / 5.6M	1/4 WATT 5%	50 PCS.

ASST. 8R Includes Resistor Assortments 1-7 (350 PCS.) **$9.95 ea.**

$5.00 Minimum Order – U.S. Funds Only
California Residents – Add 6% Sales Tax

Spec Sheets—25¢
1978 A Catalog Available—Send 41¢ stamp

NEW NAME

Jameco ELECTRONICS

PHONE ORDERS WELCOME (415) 592-8097

MAIL ORDER ELECTRONICS – WORLDWIDE
1021 HOWARD AVENUE, SAN CARLOS, CA 94070
Advertised Prices Good Thru September

J1

WIRE-WRAP KIT — WK-2-W

WRAP • STRIP • UNWRAP

- Tool for 30 AWG Wire
- Roll of 50 Ft. White or Blue 30 AWG Wire
- 50 pcs. each 1", 2", 3" & 4" lengths — pre-stripped wire

$12.95

WIRE WRAP TOOL WSU-30

WRAP • STRIP • UNWRAP - $6.95

WIRE WRAP WIRE — 30 AWG

25 ft. min. $1.25 50 ft. $1.95 100 ft. $2.95 1000 ft. $15.00
SPECIFY COLOR — White - Yellow - Red - Green - Blue - Black

WIRE DISPENSER — WD-30

- 50 ft. roll 30 AWG KYNAR wire wrap wire **$3.95 ea.**
- Cuts wire to desired length
- Strips 1" of insulation **Specify** — Blue-Yellow-White-Red

REPLACEMENT DISPENSER SPOOLS FOR WD 30
Specify blue, yellow, white or red **$1.98/spool**

EXAR

XR2206KA $14.95 Function Generator Kit (includes chip, P.C. Board and instructions)

XR2206KB $19.95 Function Generator Kit (includes all components, P.C. Board and instructions)

XR-L555 $1.50 Micro-Power version of the popular 555 Timer and directly interchangeable. Dissipates 1/15th the power and operates down to 2.7 volts. Perfect for battery operation and CMOS circuits.

XR2242CP $1.50 Precision timing circuit for generating timing pulses in **minutes, hours** and days or up to 1 year by using two. Reduces cost of time delay circuits. Basic 555 Timer with built-in 8-bit Counter.

XR205	$ 8.40			XR2556	$ 3.20
XR210	4.40	XR1488	1.39	XR2567	2.99
XR215	6.40	XR1800	3.20	XR3403	1.25
XR320	1.55	XR2206	4.40	XR4136	1.25
XR555	.39	XR2207	3.85	XR4151	2.85
XR556	.99	XR2208	5.20	XR4194	1.85
XR567CP	.99	XR2209	1.75	XR4202	3.89
XR567CT	1.25	XR2211	5.85	XR4212	1.35
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

ZENERS — DIODES — RECTIFIERS

[illegible]

SCR AND FW BRIDGE RECTIFIERS

[illegible]

TRANSISTORS

[illegible]

CAPACITOR CORNER

50 VOLT CERAMIC DISC CAPACITORS

[illegible]

100 VOLT MYLAR FILM CAPACITORS

[illegible]

±20% DIPPED TANTALUMS (SOLID) CAPACITORS

[illegible]

MINIATURE ALUMINUM ELECTROLYTIC CAPACITORS — Axial Lead / Radial Lead

[illegible]

To get overseas trade leads like this, you could open offices in 127 countries.

Or use our computer.

The U.S. Commerce Department's computer-operated Trade Opportunities Program (TOP) can supply you with immediate, continuing, specific leads tailored to your sales objectives for any of 127 countries. So if you can't be all over the world at one time, we've got the answer. Send us the coupon now and see how you can find where the live leads are living.

Secretary of Commerce
U.S. Department of Commerce, BIC-9A
Washington, D.C. 20230

Please tell me more about the Trade Opportunities Program (TOP).

Name ____________________

Title ____________________

Company ____________________

Address ____________________

City ________ State ______ Zip ______

NEW ADDITIONS TO THE **KB BOOK NOOK**

SPECIAL: PRE-PUBLICATION OFFER (orders will be filled on a first-in, first-out basis)

• HOW TO BUILD A MICROCOMPUTER—AND REALLY UNDERSTAND IT—BK7325—by Sam Creason. The electronics hobbyist who wants to build his own microcomputer system now has a practical "How-To" guidebook. Sam Creason's book is a combination technical manual and programming guide that takes the hobbyist step-by-step through the design, construction, testing and debugging of a complete microcomputer system. This book is must reading for anyone desiring a true understanding of small computer systems. $9.95.*

• PROGRAMMING IN PASCAL—BK1140—by Peter Grogono. The computer programming language PASCAL was the first language to embody in a coherent way the concepts of structured programming which had been defined by Edsger Dijkstra and C. A. R. Hoare. As such it is a landmark in the development of programming languages. PASCAL was developed by Niklaus Wirth in Zurich; it is derived from the language ALGOL 60 but is more powerful and easier to use. PASCAL is now widely accepted as a useful language that can be efficiently implemented, and as an excellent teaching tool. This book is intended for people who want to write programs in PASCAL. It does not assume knowledge of any other programming language and it is therefore suitable for an introductory course. $9.95.*

• MICROPROCESSOR LEXICON—ACRONYMS AND DEFINITIONS—BK1137—Compiled by the staff of SYBEX is a convenient reference in pocket size format. Sections include acronyms and definitions, part numbers and their definitions, S-100 signals, RS232 signals, IEEE 499 signals, microcomputer and microprocessors, JETDS summary (military) and a code conversion table. $2.95.*

• INSTANT BASIC—BK1131—by Jerald R. Brown. For the personal computer enthusiast or the user of DEC's BASIC PLUS language, here, finally, is a new book to teach you BASIC. It teaches BASIC to beginners using interesting programming ideas and applications that will be easily understood by the home computer programmer. BASIC PLUS users know that the two languages are very similar, so this book can be used by them as well. This is an "active participation" workbook, designed to be used with your home computer so you can learn by doing! Ideas are slowly introduced in a nonmathematical context so the beginner can quickly learn good programming techniques. $6.00.*

• YOUR HOME COMPUTER—BK1172—by James White, is an introduction to the world of personal microcomputing. Written in clear and understandable language, this book tells you everything you want to know about home computing and gives the computer novice a painless introduction to microcomputer technology and terminology, beginning with what computers are and how they work. This basic book requires no prior knowledge or experience in electronics or computing.

It provides information about home computer kits, guidelines for selecting and building your own microcomputer, how to use your home computer and what you can do with it, lists of computer stores, clubs, periodicals, and answers to many more of your questions about microcomputers and the jargon surrounding the personal computing scene today. $6.00.*

INSTANT SOFTWARE FROM KILOBAUD

• INSTANT SOFTWARE FROM KB The following are the first of many prerecorded programs that will be available from **Kilobaud**. Each cassette is accompanied by a booklet that includes a hard-copy version of the program as well as a description of intended use. All cassettes are available at $7.95* unless otherwise listed.

• BASIC LUNAR LANDER/INTERMEDIATE LUNAR LANDER—SP0101—by Ed Juge (for TRS-80™) Your automatic landing system has malfunctioned and you are forced to make a manual landing. Taking control at 15,000 feet, the pilot's success requires precise manipulation of the lunar lander's retro-rockets. After mastering the basic game, the able commander will be taxed by the difficult intermediate program. $7.95.*

• PERSONAL WEIGHT CONTROL/BIORHYTHMS—SP0105—by Microcosm (for the PET™) Here's a program that will let you chart any individual's biorhythm cycle, showing physical, emotional, and intellectual highs and lows. If your neighbors find out you've got it, they will never leave you in peace. **PERSONAL WEIGHT CONTROL** is a must for the diet-conscious. Once your ideal weight has been determined, the program will calculate the correct daily caloric intake necessary to reach your goal over a certain time period. $7.95.*

• SPACE TREK II—SP0102—by Ed Juge (for the TRS-80™) This Space Trek game will challenge the most skillful and careful player. As commander of the Enterprise, you must defend the galaxy against enemy Klingon warships using your arsenal of phasers and photon torpedoes to best advantage. The Enterprise is also equipped with impulse power for moving within the quadrant, and warp power for jumping from quadrant to quadrant. $7.95.*

• MORTGAGE WITH PREPAYMENT OPTION/FINANCIER—SP0106—by Microcosm (for the PET™) Before you go to the bank, calculate your mortgage payment schedule and find out what you can afford. You will learn how much prepayments save, and can easily compare the true costs of variations in interest. **FINANCIER** is actually three programs in one and is guaranteed to make you a financial wizard. In part I, you'll learn to tell which investments will pay off. Part II calculates annual depreciation schedules, providing rates, amounts, and salvage values, which is great when tax time rolls around. The final section figures costs of borrowing, payment terms, and remaining balances on loans. $7.95.*

• BEGINNER BACKGAMMON/KENO—SP0104—by Fred Schilds and Ed Juge (for the TRS-80™) You will never be in need of a backgammon companion again! Schilds's program is designed for the newcomer to the game, but the computer's steady play will challenge the careless expert. **KENO** is a popular Las Vegas gambling game. As in many games of chance, the odds are against you, but when you win you can win big. $7.95.*

* Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to: Kilobaud Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information. For orders under $10.00, add $1.00 shipping and handling.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

KB BOOK NOOK

● HOBBY COMPUTERS ARE HERE!—BK7322—If you want to come up to speed on how computers work . . . hardware and software . . . this is an excellent book. It starts with the fundamentals and explains the circuits, the basics of programming, along with a couple of TVT construction projects, ASCII-Baudot, etc. This book has the highest recommendations as a teaching aid for newcomers. $4.95.*

● THE NEW HOBBY COMPUTERS—BK7340—This book takes it from where "**Hobby Computers Are Here!**" leaves off, with chapters on Large Scale Integration, how to choose a microprocessor chip, an introduction to programming, low cost I/O for a computer, computer arithmetic, checking memory boards, a Baudot monitor/editor system, an audible logic probe for finding those tough problems, a ham's computer, a computer QSO machine . . . and much, much more! $4.95*

● INTRODUCTION TO MICROPROCESSORS—BK1032—by Charles Rockwell of MICROLOG is an ideal reference for the individual desiring to understand the hardware aspects of microprocessor systems. Describes the hardware details of computer devices in terms the beginner can understand, instead of treating the micro chip as a "black box." Specific systems are not described and programming is only briefly discussed. $17.50 US and Canada, $20 elsewhere.*

● BASIC NEW 2ND EDITION—BK1081—by Bob Albrecht. Self-teaching guide to the computer language you will need to know for use with your microcomputer. This is one of the easiest ways to learn computer programming. $4.95.*

● SCIENTIFIC RESEARCH INSTRUMENTS' BASIC SOFTWARE LIBRARY is a complete do-it-yourself kit. Written in everybody's BASIC immediately executable in ANY computer with 4K, no other peripherals needed. **Volume I** contains business and recreational programs and is 300 pages. **Volume II** is 260 pages and contains math, engineering, statistics and plotting programs. **Volume III** contains money managing, advanced business programs such as billing, A/R, inventory, payroll, etc. **Volume IV** contains general purpose programs like loans, rates, retirement, plus games: Poker, Enterprise (take charge while Capt. Kirk is away), Football and more! **Volume V** is filled with experimenter's programs including games, pictures and misc. problems like "logic." Volume I—LB1002—& Volume II—LB1003—$24.95* each, Volume III—LB1004—$39.95*, Volume IV—LB1005—& Volume V—LB1006—$9.95* each, Volume VI—LB1007—$49.95*, Volume VII—LB1008—$39.95.*

● MICROPROCESSOR INTERFACING TECHNIQUES—BK1037—by Austin Lesea & Rodnay Zaks will teach you how to interconnect a complete system and interface it to all the usual peripherals. It covers hardware and software skills and techniques, including the use and design of model buses such as the IEEE 488 or S100. $9.95*

● TTL COOKBOOK—BK1063—by Donald Lancaster. Explains what TTL is, how it works, and how to use it. Discusses practical applications, such as a digital counter and display system, events counter, electronic stopwatch, digital voltmeter, and a digital tachometer. $8.95.

● CMOS COOKBOOK—BK1011—by Don Lancaster. Details the application of CMOS, the low power logic family suitable for most applications presently dominated by TTL. Required reading for every serious digital experimenter! $9.95.*

● TVT COOKBOOK—BK1064—by Don Lancaster, describes the use of a standard television receiver as a microprocessor CRT terminal. Explains and describes character generation, cursor control and interface information in typical, easy-to-understand Lancaster style. $9.95.

● THE "COMPULATOR" BOOK—BUILDING SUPER CALCULATORS & MINICOMPUTER HARDWARE WITH CALCULATOR CHIPS—BK1012—by R. P. Haviland, provides ideas, design info and printed circuit boards for calculator chip projects, measure time, tie in with a Teletype to create a virtually infinite memory system, and countless other functions. $7.95.*

● BASIC COMPUTER GAMES—BK1074—Okay, so once you get your computer and are running in BASIC, then what? Then you need some programs in BASIC, that's what. This book has 101 games for you, from very simple to real buggers. You get the games, a description of the games, the listing to put in your computer and a sample run to show you how they work. Fun. Any one game will be worth more than the price of the book for the fun you and your family will have with it. $7.50.*

● WHAT TO DO AFTER YOU HIT RETURN—BK1071—PCC's first book of computer games . . . 48 different computer games you can play in BASIC . . . programs, descriptions, many illustrations. Lunar Landing, Hammurabi, King, Civel 2, Qubic 5, Taxman, Star Trek, Crash, Market, etc. $8.00.*

● KILOBAUD CODING SHEETS Kilobaud's dozen or so programmers wouldn't try to work without these handy scratch pads which help prevent the little errors that can cost hours and hours of programming time. Available for programming is **Assembly/Machine Language** (PD1001) which has columns for address, instruction (3 bytes), source code (label, op code, operand), and comments; and for **BASIC** (PD1002) which is 72 columns wide. 50 sheets to a pad. $2.39.*

* Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to:
Kilobaud Book Department ● Peterborough NH 03458. Be sure to include check or detailed credit card information.
For orders under $10.00, add $1.00 shipping and handling.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

WHO'S BEHIND THE kilobaud ?

WHO'S THAT CLOWN BEHIND THE KILOBAUD?

One hint . . . no, it's not the editor of one of the other hobby computer magazines . . . this chap is the designer of one of (if not the) best selling RAM boards in history.

LAST MONTH'S MYSTERY READER

Yes, Virginia, there really is a Bill Godbout, and he reads KILOBAUD . . . and loves it. Look at that smile. Bill has been reading his ad and is thinking of the truckloads of mail orders it will be bringing him. If you missed guessing Bill last month, then you will win the Dr. Watson award . . . half his name was showing over his right ear.

Subscribe Today!
For Instant Subscriptions Call Our Toll Free Number
1 (800) 258-5473
Have your credit card handy!
Kilobaud • Peterborough NH 03458

MOVING?

Let us know 8 weeks in advance so that you won't miss a single issue of *Kilobaud*.
Attach old label where indicated and print new address in space provided. Also include your mailing label whenever you write concerning your subscription. It helps us serve you promptly.

- ☐ Address change only
- ☐ Extend subscription
- ☐ Enter new subscription
- ☐ 1 year $18.00
- ☐ Payment enclosed (1 extra BONUS issue)
- ☐ Bill me later

If you have no label handy, print OLD address here.

AFFIX LABEL

Name__________ Call______
Address__________
City______ State______ Zip______

print NEW address here:

Name__________ Call______
Address__________
City______ State______ Zip______

• KILOBAUD • PETERBOROUGH NH 03458 •

1977 KB BACK ISSUES

All issues subject to availability!
Issue #1—$5.00; #2 thru #12—$3 ea.; #13 & #14—$10 ea.
Please send Kilobaud back issues:

___ issues JAN 77-$5	___ issues MAY 77—$3	___ issues SEPT 77—$3
___ issues FEB 77—$3	___ issues JUNE 77—$3	___ issues OCT 77—$3
___ issues MAR 77—$3	___ issues JUL 77—$3	___ issues NOV 77—$3
___ issues APR 77—$3	___ issues AUG 77—$3	___ issues DEC 77—$3
		___ issues JAN 78—$10
		___ issues FEB 78—$10

TOTAL: ________

Have credit card # handy and call
TOLL FREE (800) 258-5473
Kilobaud • Peterborough NH 03458

KB/3/78

Add $1.00 shipping & handling for all orders under $10.00.

☐ BankAmericard/Visa ☐ Master Charge ☐ American Express

Card # ____________ Expiration date ______

Signature ____________

Name ____________

Address ____________

City ____________ State ____ Zip ______

FULL SIZE FLOPPY DISK $995 COMPLETE!

DISCUS I™ full-size floppy disk system is an overnight success . . . because it's delivered so complete you can have it running in a single evening.

For just $995, it's a complete memory system. Complete with all hardware and software. Completely assembled. Completely interfaced. And tested as a complete system.

And you can not only solve your memory shortage faster, you can solve it longer . . . because DISCUS I™ is a full-size floppy system with 3 times the storage and 5 times the speed of mini-floppies.

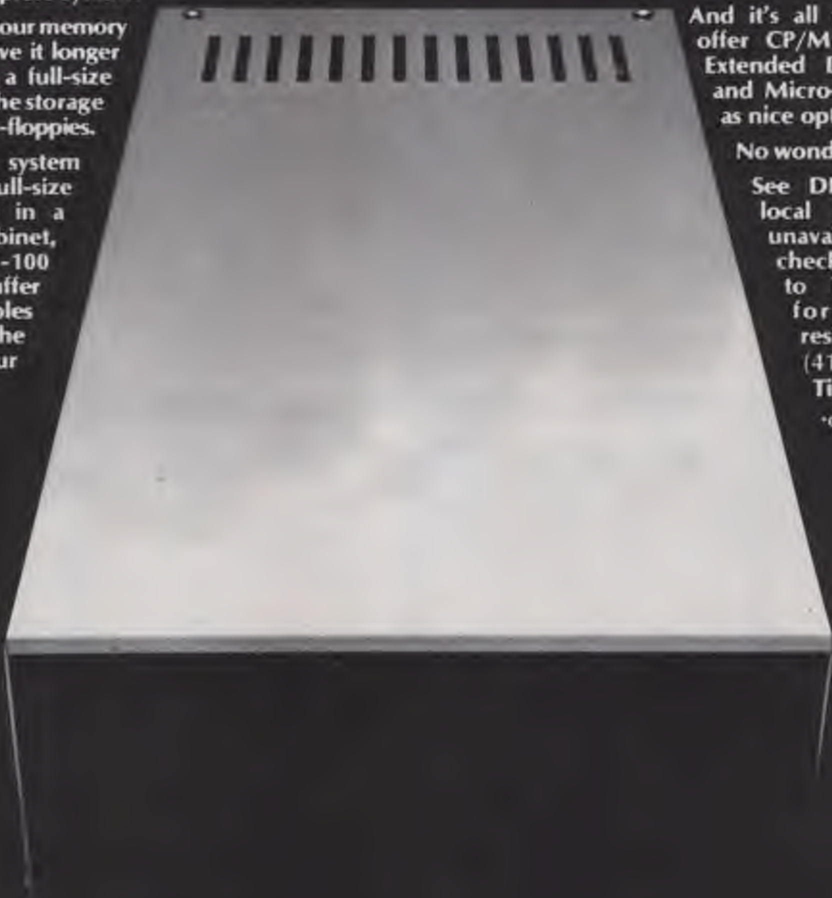
Your $995 DISCUS I™ system includes a Shugart 800R full-size drive with power supply in a handsome freestanding cabinet, our 8-drive capacity S-100 controller with on-board buffer and serial interface, all cables and connectors, and all the software you need. Your software library includes DOS, text editor, 8080 assembler (all integrated in DISK/ATE™), our BASIC-V™ advanced virtual disk BASIC able to handle a wide variety of data formats and address up to 2 megabytes and patches for CP/M*. And it's all interfaced to your controller's serial I/O port to avoid I/O guesswork.

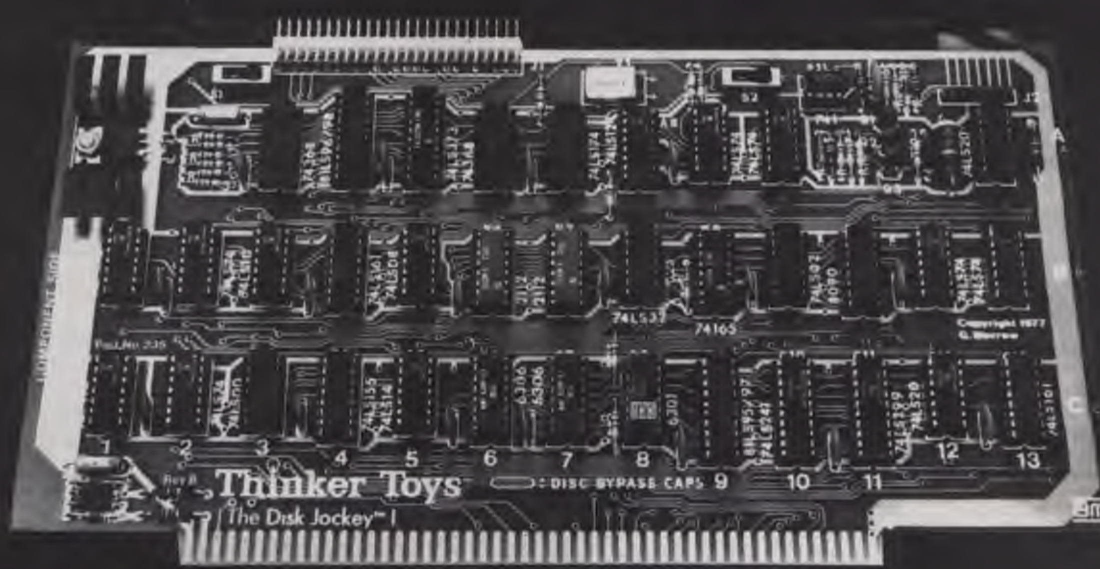
And it's all yours for $995. We even offer CP/M for just $70, Micro-Soft Extended Disk Basic for just $199 and Micro-Soft Fortran for just $349 as nice options to add to your library.

No wonder it's an overnight success!

See DISCUS I™ today at your local computer shop. Or if unavailable locally, send your check or money order direct to Thinker Toys™ (add $7 for handling; California residents add tax). Or call (415)-524-5317, 10-5 Pacific Time.

*CP/M is a trademark of Digital Research.





A product of Morrow's Micro-Stuff for

Thinker Toys™

1201 10th Street
Berkeley, CA 94710

kilobaud

index of advertisers

Send for your free product catalogs

Return this card to receive full information on the many fine products advertised in this issue. Simply circle the Reader Service # below for each advertised product. The codes appear in the index of advertisers on this page, and near the company's logo on the ad pages. Please provide a full address and zip code. (Limit: 25 requests)

☐ **Subscriber** ☐ **Newsstand**

A38	C34	C81	D28	G4	J12	M51	O6	P36	S2	S60	T38
A48	C46	C82	D30	G15	K14	M54	08	P37	S6	S61	W13
B14	C64	C85	D32	H20	M17	N9	09	P38	S46	T1	W16
B28	C65	C86	E29	I9	M30	N12	O10	P39	S51	T11	W20
B33	C70	C87	E30	I13	M31	N13	P7	Q3	S52	T13	73
B34	C73	D13	E34	I21	M32	O1	P9	R8	S54	T26	
C28	C77	D20	E35	I24	M44	O2	P21	R12	S55	T28	kilobaud
C31	C79	D26	F4	J1	M49	O3	P24	R17	S56	T36	
C33	C80	D27	F8	J4	M50	O5	P35	R23	S59	T37	

This month's ARTICLE WINNER (title page number): ____________

Name ________________________________

Address ________________________________

City ____________________ State ______ Zip ____________

9/78

FIRST CLASS
Permit No. 17
Peterborough NH 03458

BUSINESS REPLY MAIL

No postage stamp necessary if mailed in the United States

Postage will be paid by

kilobaud

44 Main Street
Peterborough NH 03458

Attn: Mail Order

Send for your free product catalogs

Return this card to receive full information on the many fine products advertised in this issue. Simply circle the Reader Service # below for each advertised product. The codes appear in the index of advertisers on this page, and near the company's logo on the ad pages. Please provide a full address and zip code. (Limit: 25 requests)

☐ **Subscriber** ☐ **Newsstand**

A38	C34	C81	D28	G4	J12	M51	O6	P36	S2	S60	T38
A48	C46	C82	D30	G15	K14	M54	08	P37	S6	S61	W13
B14	C64	C85	D32	H20	M17	N9	09	P38	S46	T1	W16
B28	C65	C86	E29	I9	M30	N12	O10	P39	S51	T11	W20
B33	C70	C87	E30	I13	M31	N13	P7	Q3	S52	T13	73
B34	C73	D13	E34	I21	M32	O1	P9	R8	S54	T26	
C28	C77	D20	E35	I24	M44	O2	P21	R12	S55	T28	kilobaud
C31	C79	D26	F4	J1	M49	O3	P24	R17	S56	T36	
C33	C80	D27	F8	J4	M50	O5	P35	R23	S59	T37	

This month's ARTICLE WINNER (title page number): ____________

Name ________________________________

Address ________________________________

City ____________________ State ______ Zip ____________

9/78

FIRST CLASS
Permit No. 17
Peterborough NH 03458

BUSINESS REPLY MAIL

No postage stamp necessary if mailed in the United States

Postage will be paid by

kilobaud

Peterborough NH 03458

Attn: Reader Service

SUBSCRIBE! ☐ Kilobaud ☐ 73 Magazine

☐ 1 year @ $18* ☐ 3 years @ $36* ☐ Life Sub @ $195**

Renewal—Expira. date______ ☐ New subscription

*U.S. & Canada only. (U.S. currency, please) **U.S. only

Please send me the following Kilobaud products: PLEASE PRINT OR TYPE

Qty.	Catalog #	Description	Unit Price	Total
		Add $1.00 shipping & handling for orders under $10.00		
			Total:	

Coupon expires in 60 days.

Enclosed $________ ☐ Check ☐ Money order

Bill my: ☐ American Express ☐ BankAmericard/Visa ☐ Master Charge

Credit card #________________________________

Expiration date ________ Signature ________________________

Name ________________________________

Address ________________________________

City ____________________ State _____ Zip ____________ 9/78

FIRST CLASS
Permit No. 17
Peterborough NH 03458

BUSINESS REPLY MAIL

No postage stamp necessary if mailed in the United States

Postage will be paid by

kilobaud

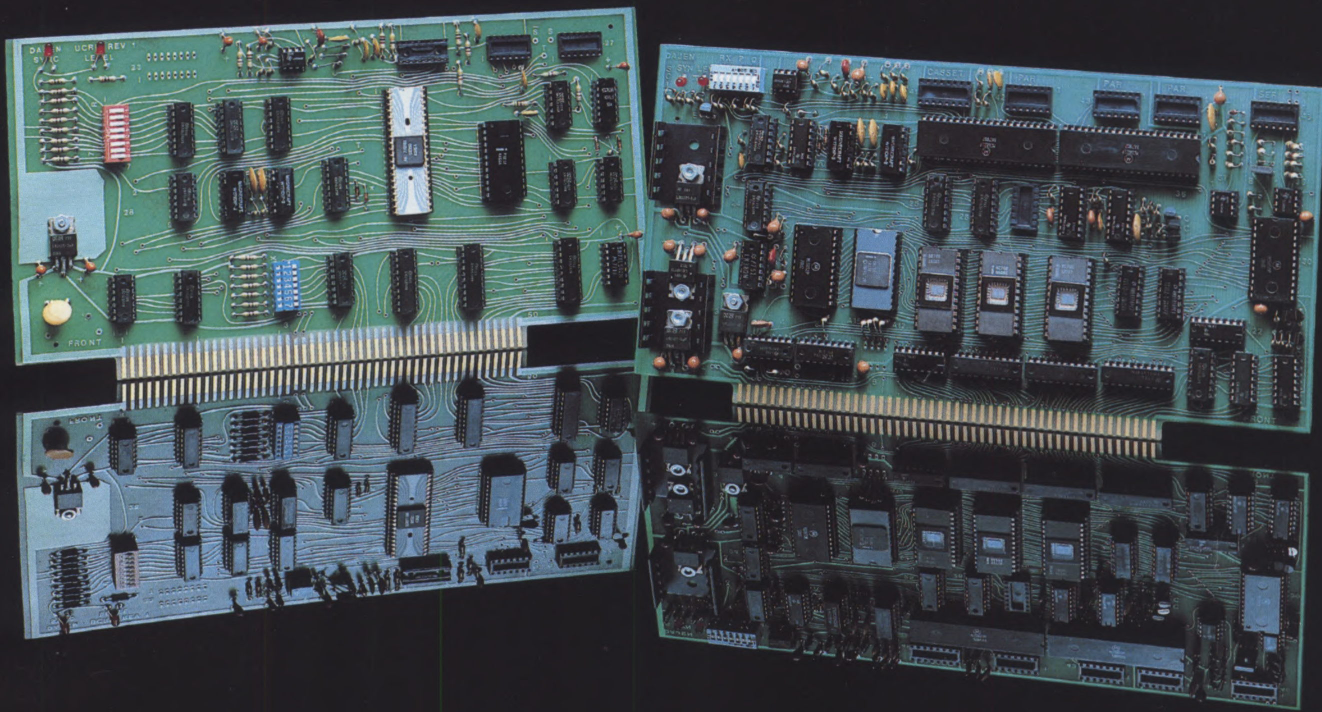
Peterborough NH 03458

Attn: Mail Order

kilobaud

books, etc.

BK1074 BASIC Computer Games—$7.50
BK1081 BASIC New 2nd Editio[n] $4.95
BK1011 CMOS Cookbook—$9.[95]
BK7322 Hobby Computers Are Here!—$4.95
BK7325 How To Build A Microcomputer—And Really Unde[r]stand It—$9.95
BK1131 Instant BASIC—$6.00
Instant Software From Kilobau[d]
SP0101 BASIC Lunar Lander termediate Lunar Lander—$7.95
SP0104 Beginner Backgamm[on] Keno—$7.95
SP0106 Mortgage With Prepa[y]ment Option/Financier—$[7.95]
SP0105 Personal Weight Co[n]trol/Biorhythms—$7.95
SP0102 Space Trek—$7.95
BK1032 Introduction To Microp[ro]cessors—$17.50 USA; $20 E[lse]where
Kilobaud Coding Sheets
PD1001 Assembly/Machine L[an]guage—$2.39
PD1002 BASIC—$2.39
BK1037 Microprocessor Interfac[ing] Techniques—$9.95
BK1137 Microprocessor Lexico[n] Acronyms And Definitions—$2.95
BK1140 Programming In Pasca[l] $9.95
Scientific Research Instrument[s] BASIC Software Library
LB1002 Vol. I—$24.95, LB100[3] Vol. II—$24.95, LB1004 Vol. I[II] $39.95, LB1005 Vol. IV—$9.9[5] LB1006 Vol. V—$9.95, LB100[7] Vol. VI—$49.95, LB1008 Vol. —$39.95
BK1012 The "Compulator" Boo[k] Building Super Calculators & Minicomputer Hardware Wit[h] Calculator Chips—$7.95
BK7340 The New Hobby Comp[u]ters—$4.95
BK1063 TTL Cookbook—$8.95
BK1064 TVT Cookbook—$9.95
BK1071 What To Do After You [Hit] Return—$8.00
BK1172 Your Home Computer—$6.00



Two To The Byte Power

THE I/O POWER YOU NEED:

Our Universal Cassette Recorder Interface speaks to cassettes in three languages — biphase (Tarbell), CUTS, and Kansas City — at speeds from 600 to 40,000 baud. In addition, on-board relays control two recorders. A keyboard or other external device can speak to your CPU via the UCRI's parallel input port. S-100 compatible.

Teletek's System Central Interface (SCI), the most powerful interface on the market, gives you all of the features of the UCRI plus a 2708 programmer, serial port, three parallel ports, on board EPROM's, reset-jump, RAM, monitor program, and more.

For unexcelled Byte Power, call or write Ann Roberts, Department 36, at (916) 351-0535.

TELETEK: QUALITY PRODUCTS, A DECADE OF SERVICE

SEE YOUR DEALER
OEM CONTRACTS AVAILABLE

TELETEK
11505B Douglas Rd.
Rancho Cordova, CA 95670